

# POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria del Cinema e  
dei Mezzi di Comunicazione



Tesi di Laurea Magistrale

## Sviluppo del progetto d'animazione 3D Reverie Dawnfall tramite tecniche di shading e lighting non fotorealistico

Relatore

Prof. Riccardo ANTONINO

Candidato

Eugenio NICOLOSI

Dicembre 2023



# Sommario

L'ambito dell'animazione e della grafica 3D ha conosciuto una notevole evoluzione negli ultimi decenni, con una crescente enfasi sulla creazione di opere visivamente sorprendenti e artisticamente accattivanti, che simulino non solo ambienti fotorealistici ma anche estetiche tipiche dell'animazione tradizionale. Uno degli aspetti fondamentali che contribuisce a raggiungere tali obiettivi di rendering (processo mediante il quale un'immagine tridimensionale viene trasformata in una rappresentazione bidimensionale) è quello di appositi shader. L'universo dello shading non fotorealistico rappresenta una dimensione affascinante e creativa all'interno della computer grafica, dove l'obiettivo principale non è tanto la riproduzione fedele della realtà, ma piuttosto l'esplorazione di stili artistici unici e accattivanti. Nel contesto dello sviluppo serie animata *Reverie Dawnfall*, questa tesi si propone di illustrare il lavoro effettuato sullo shading e contemporaneamente analizzare lo sviluppo del progetto sin dalla fase di riproduzione. La lavorazione del progetto ha infatti richiesto un lavoro preliminare che verrà illustrato, a cui è seguito uno sviluppo dal punto di vista creazione degli asset, illuminazione di ambienti 3D e shading dei modelli.

L'obiettivo principale di questo studio è di fornire un'estetica distintiva e non fotorealistica che affondasse le sue radici nel mondo del fumetto tramite il motore di render Eevee di Blender. Lo studio di diverse opere che si riconducevano a questo stile ha reso possibile che venisse realizzato uno shader di tipo toon che rielaborasse quello creato in precedenza per il progetto. Questa rielaborazione ha significato anche la sperimentazione di diverse tecniche che non hanno contribuito all'aspetto finale ma di cui è interessante analizzarne le potenzialità. Con lo scopo di creare dettagli light indipendenti e fornire un look tipico delle tecniche tradizionali 2D, sono state analizzate tecniche come quella del camera projection, che serve a proiettare una texture su un oggetto dal punto di vista della camera, o normal paint strokes, tecnica per la creazione di un effetto dipinto con modifica delle normali, per poi effettuare una sperimentazione più mirata su gli aspetti principali dello shader. Gli elementi di sviluppo principali sono stati effettuati per creare o migliorare gli effetti di halftoning, retino a righe e outline.

Prima dello studio sullo shading è stata effettuata un'analisi sul software di

grafica 3D Blender ed è stato eseguito un confronto delle caratteristiche dei motori di rendering messi a disposizione Cycles e Eevee, con un'attenzione su quest'ultimo. Inoltre è stato esplorato il processo di sviluppo che ha preceduto la realizzazione della scena dell'episodio pilota della serie presa in considerazione, partendo dalla fase di preproduzione fino a quella di produzione. Questo lavoro preliminare ha coinvolto un'analisi della sceneggiatura, attraverso lo spoglio, per proseguire con la definizione di uno storyboard e animatic. Alla fine di questo processo sono stati definiti i vari task necessari alla realizzazione del progetto. Inoltre la tesi illustra il lavoro effettuato in fase di lighting per raggiungere l'aspetto artistico definito in fase di preproduzione. Successivamente, sono stati esaminati i diversi stadi del processo di definizione dell'illuminazione all'interno dell'ambiente tridimensionale, passando per un'analisi delle potenzialità di lighting di Eevee all'interno del software Blender.

In ultima analisi verranno mostrati i risultati del lavoro effettuato e analizzato le caratteristiche dello shader e possibili punti di sviluppo futuri.

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Reverie Dawnfall . . . . .	1
1.2	Sviluppo produttivo e obiettivi . . . . .	3
<b>2</b>	<b>Stato dell'arte</b>	<b>5</b>
2.1	Spider-Man: Into the Spider-Verse e Spider-Man: Across the Spider-Verse . . . . .	5
2.2	Love, Death & Robots . . . . .	7
2.3	Arcane . . . . .	8
<b>3</b>	<b>Tecniche e strumenti</b>	<b>10</b>
3.1	Blender . . . . .	10
3.2	Motore di rendering . . . . .	11
3.3	Differenze fra Cycles e Eevee . . . . .	12
3.3.1	Illuminazione globale . . . . .	12
3.3.2	Riflessioni . . . . .	13
3.3.3	Rifrazioni . . . . .	14
3.3.4	Trasparenza . . . . .	15
3.3.5	Ambient occlusion . . . . .	15
3.3.6	Ombre . . . . .	16
3.3.7	Luce volumetrica . . . . .	16
3.3.8	Subsurface scattering . . . . .	17
3.3.9	Profondità di campo . . . . .	18
3.3.10	Motion blur - Eevee . . . . .	19
3.3.11	Bloom . . . . .	20
<b>4</b>	<b>Produzione</b>	<b>21</b>
4.1	Preproduzione . . . . .	21
4.2	Spoglio sceneggiatura . . . . .	22
4.3	Storyboard . . . . .	24
4.4	Animatic . . . . .	26

4.5	Workflow . . . . .	27
<b>5</b>	<b>Shading</b>	<b>30</b>
5.1	Obbiettivi . . . . .	30
5.2	Ambiente shading di Blender . . . . .	31
5.3	Shading procedurale . . . . .	33
5.4	Sistemi di coordinate . . . . .	33
5.4.1	Generated . . . . .	33
5.4.2	Normal . . . . .	34
5.4.3	UV . . . . .	35
5.4.4	Object . . . . .	36
5.4.5	Camera . . . . .	37
5.4.6	Window . . . . .	38
5.4.7	Reflection . . . . .	39
5.4.8	Tangent . . . . .	40
5.4.9	Position(Location) . . . . .	40
5.4.10	View vector . . . . .	41
5.5	Setup dello shader . . . . .	42
5.6	Outline . . . . .	42
5.6.1	Metodo Freestyle . . . . .	43
5.6.2	Line art modifier . . . . .	44
5.6.3	Inverted hull . . . . .	45
5.6.4	Outline dinamico . . . . .	46
5.7	Pattern a esagono . . . . .	48
5.8	Halftoning . . . . .	50
5.9	Retino a righe . . . . .	52
5.10	Ologramma . . . . .	54
5.11	Prove . . . . .	55
5.11.1	Normal paint strokes . . . . .	55
5.11.2	Camera projection . . . . .	56
<b>6</b>	<b>Lighting</b>	<b>58</b>
6.1	Teorie e tecniche dell'illuminazione . . . . .	58
6.2	Ambiente Blender . . . . .	61
6.3	Reference . . . . .	64
6.4	Setup luci nella scena . . . . .	64
<b>7</b>	<b>Riultati e conclusioni</b>	<b>67</b>
7.1	Risultati . . . . .	67
7.2	Conclusioni . . . . .	69
	<b>Bibliografia</b>	<b>71</b>





# Capitolo 1

## Introduzione

Negli ultimi anni lo stato delle tecniche d'animazione ha subito una grande evoluzione, con innovazioni tecnologiche e artistiche che hanno rivoluzionato l'industria. L'animazione 2D continua ad avere una sua rilevanza, ma ora si affianca a tecniche digitali che semplificano il processo di animazione. L'animazione 3D, alimentata da potenti software di modellazione e rendering, ha rivoluzionato l'industria cinematografica, consentendo la creazione di mondi virtuali sempre più realistici. Questa evoluzione delle potenzialità dei software di modellazione e animazione ha reso non solo possibile lo sviluppo di opere animate in digitale nella direzione di simulazioni più realistiche ma anche verso scelte artistiche più peculiari e proprie del mondo dell'animazione e illustrazione 2D.

Le nuove potenzialità tecnologiche ed espressive hanno fatto sì che anche studi di produzione più indipendenti avessero la possibilità di realizzare progetti animati con risultati grafici di alta qualità e permettendo il proliferare di scelte stilistiche originali nel mondo della produzione cinematografica e delle serie tv. Una piccola produzione ha i mezzi per sviluppare dall'inizio alla fine un progetto d'animazione con tutte le competenze, che vanno dallo sviluppo della storia alla realizzazione effettiva del prodotto audiovisivo.

In questo contesto si inserisce la storia produttiva di *Reverie Dawnfall*, una serie animata prodotta da Robin Studio, che cerca di sfruttare le nuove tecniche per sviluppare un progetto originale dal punto di vista creativo ed estetico.

### 1.1 Reverie Dawnfall

*Reverie Dawnfall*(1.1) è un progetto d'animazione in 3D fantascientifica in ambientazione distopica cyberpunk, che racconta storia di Nadya e un gruppo di ragazzi nati con disabilità fisiche o disturbi neuropsichiatrici a causa del gravissimo inquinamento([1]).



**Figura 1.1:** Reverie cover

La serie è ambientata in un pianeta in cui gli umani devono fare i conti con delle condizioni ambientali estreme e che ruota sincronicamente rispetto alla sua stella, il che significa che un emisfero è costantemente in ombra mentre l'altro è sempre illuminato. L'unico modo per far sì che gli umani possano sopravvivere a queste condizioni proibitive è di risiedere nella zona situata lungo la linea di demarcazione fra queste due metà. Gli uomini sono costretti a vivere in una città-alveare inserita dentro un gigantesca cupola che li separa dall'ambiente ostile esterno e che data la posizione geografica all'interno del pianeta si trova in un costante stato di luce crepuscolare che fornisce una illuminazione caratteristica a tutta la serie. I toni freddi della città ipertecnologica cyberpunk si fondono a degli sfondi caldi tipici delle storie post apocalittiche come Mad Max.(1.2)



**Figura 1.2:** Concept art

In questo pianeta tutti nascono con delle disabilità che forniscono degli svantaggi ma in alcuni casi anche dei poteri a chi ne è affetto. Tutti vengono cresciuti senza una famiglia vera e propria e in un mondo in cui le norme della società sono diverse dalla nostra. In una società in cui ognuno è imperfetto Peregrine, che è nato geneticamente puro, manovra i destini di questa società. In questo mondo governato da un'autorità totalitari il nostro gruppo di ragazzi si domanda "e se questo non fosse il mondo giusto?"

## 1.2 Sviluppo produttivo e obiettivi

Reverie Dawnfall ha una storia produttiva che parte dal 2017 quando venne creata all'interno dello studio creativo torinese Robin Studio ma che ha vissuto più fasi di sviluppo, compreso il lavoro di altri tesisti del Politecnico di Torino. Per prima cosa è stata definita la realizzazione del concept della serie con la realizzazione di documenti di riproduzione come concept art, character sheet e stesura di alcune sceneggiature ad opera di Mark Gore. Nella fase iniziale si è cercato di fornire delle linee guida sul lato della lore e dell'estetica della serie per fornire delle fondamenta da cui sviluppare il resto.

L'idea originale era quella di sviluppare un teaser trailer della durata di un minuto per raccogliere i fondi necessari alla realizzazione dell'episodio pilota. Questo teaser è stato realizzato grazie anche al lavoro di un team di precedenti tesisti che hanno portato avanti il progetto. Dopo la realizzazione di questo teaser è stato deciso da Robin Studio di realizzare direttamente delle scene presenti nell'episodio pilota da usare come reel per possibili bandi e opportunità di produzione. Il lavoro di questa tesi si concentra dunque sullo sviluppo di una scena del primo episodio insieme alla collaborazione di altri tesisti.

Il team di lavoro precedente ha svolto un lavoro riguardo la realizzazione di ambienti(1.3), personaggi, animazione e shader appositi. All'inizio della lavorazione di questo progetto di tesi è stato fornito al team di lavoro attuale i seguenti asset:

- Concept art di diversi ambienti e personaggi(incluso il night club)
- Modelli di alcuni personaggi
- Shader
- Ambiente 3D della stanza di Nadya
- Sceneggiature delle prime quattro puntate

Dopo aver analizzato la sceneggiatura con il professore Antonino si è deciso di realizzare per primo una scena dell'episodio pilota ambientata in un night club.



**Figura 1.3:** Stanza Nadya

Partendo dagli asset già realizzati, oltre a realizzare quelli nuovi, si è deciso di sviluppare un nuovo shader che partisse dal lavoro di quello precedente.

L'obiettivo del progetto è quello di sviluppare un prodotto che possa rappresentare il carattere della serie sia da un punto di vista di storytelling che di estetica. Quindi si è deciso realizzare una scena in cui ci fossero tanti personaggi presenti, con un'ambientazione che desse un'idea del carattere della serie e di sviluppare un look che fosse distintivo.

All'interno di una lavorazione di uno studio indipendente, si ci è dovuti quindi prima confrontare con una fase di sperimentazione in cui si è ricercato quale fosse la tecnica più adatta da utilizzare per realizzare l'aspetto estetico desiderato. Una volta superata questa fase si è dovuto cominciare la produzione effettiva della scena in considerazione, anche in vista del lavoro futuro da parte dei tesisti che avrebbero finito il resto del lavoro.

## Capitolo 2

# Stato dell'arte

Negli ultimi anni il mondo dell'animazione 3D ha visto un'evoluzione delle tecniche di realizzazione verso diverse direttrici. Una è quella che punta verso uno stile più fotorealistico, verso una ricerca del dettaglio e delle simulazioni più realistiche. Un'altra direzione è quella che ricerca uno stile artistico stilizzato, che affonda le sue radici nel mondo dell'animazione tradizionale e del mondo dell'illustrazione grafica, come quello dei fumetti. Quest'ultima corrente di prodotti d'animazione ha subito un particolare impulso negli ultimi anni, con il proliferare di film e serie tv che hanno dato nuova linfa allo stile.

Durante la fase di pre produzione si è dovuto fare un lavoro di ricerca dei prodotti audiovisivi che più si avvicinassero alla resa grafica che Robin studio voleva dare all'animazione. Attraverso uno studio dei film e serie tv con uno stile grafico stilizzato si è ricercato quali fossero le tecniche e i workflow utilizzati, in modo tale da capire quali fossero quelli da adottare o adattare a un tipo di produzione a basso budget.

### 2.1 Spider-Man: Into the Spider-Verse e Spider-Man: Across the Spider-Verse

"Spider-Man: Into the Spider-Verse"(2.1) è un film d'animazione che ha rivoluzionato il genere supereroico e non. Diretto da Peter Ramsey, Rodney Rothman e Bob Persichetti e prodotto dalla Columbia Pictures e dalla Sony Pictures Animation, il film è stato accolto con entusiasmo dalla critica e dal pubblico per la sua innovativa estetica e la profonda narrazione. Nel 2018 (data di uscita del film) è stato sicuramente visto come un film che potesse dare nuova linfa ai film d'animazione, prendendo ispirazione dalle tecniche di animazione più tradizionali.

Lo stile artistico da cui più attinge il film è quello del fumetto da cui prende alcune soluzioni([2]). Una di quelle più visibili è la scelta di usare l'halftoning, una

tecnica che permette di simulare sfumature di colori e gradienti attraverso l'uso di punti invece di un uso del colore più continuo. Un'altra tecnica utilizzata e molto riconoscibile è quella dell'uso dell'hatching, ovvero delle linee oblique che servono a creare o enfatizzare le ombre. Diversi dettagli come gli outline ed espressioni facciali sono stati disegnati da animatori in stile 2D. Una scelta artistica peculiare



**Figura 2.1:** Spider-Man: Into the Spider-Verse

è stata quella di usare un elemento stilistico dei primi anni del fumetto, ovvero il disallineamento dei colori in fase di stampa, che era considerato un errore, come una soluzione per una caratteristica del medium filmico, ovvero il fuori fuoco. Gli elementi fuori fuoco sono infatti gestiti con disallineando le figure e i colori per dare un senso di sfocato. Fra le tecniche prese dall'animazione tradizionale invece ci sono il passo due e il motion blur creato con lo snear. Il passo due consiste nel creare un'animazione usando lo stesso frame per due frame del film, realizzando un movimento che è meno fluido e più tradizionale. Questa tecnica è stata utilizzata inoltre in modo differente per sottolineare delle caratteristiche di alcuni personaggi in base a quanto fluido dovesse sembrare il loro movimento. La tecnica dello snear invece è stata utilizzata in sostituzione al motion blur, sovrapponendo uno stesso elemento nello stesso frame in momenti temporali diversi per creare un effetto scia.

La maggior parte di queste tecniche sono state realizzate con l'aiuto di animatori 2D che hanno lavorato frame per frame e disegnando digitalmente sopra il render del computer, creando dei layer da applicare al prodotto finale e richiedendo l'impiego di quasi 177 animatori, quasi il doppio del numero impiegato in produzioni di budget simile. Ciò implica che questo modello di produzione, per quanto sia un punto di riferimento dal punto di vista stilistico non possa rappresentare un punto di riferimento del punto di vista produttivo. Infatti, trattandosi la nostra di una produzione a basso budget, si è cercato di trovare soluzioni che fossero più automatizzate e implementate con l'uso più possibile di shader.

## 2.2 Love, Death & Robots

Love, Death & Robots è una serie antologica d'animazione disponibile su Netflix. Creata da Tim Miller e David Fincher, la serie offre una collezione di storie animate di diversi generi, stili e ambientazioni, unite da tematiche comuni di fantascienza, horror e azione.

La serie si distingue per la sua eclettica gamma di stili di animazione, che spaziano dalla computer graphics all'animazione tradizionale, dal fotorealismo all'estetica da fumetto. Questa diversità di stili visivi è uno dei punti di forza di "Love, Death & Robots" e dimostra la versatilità e la creatività dell'animazione. Questo compendio di stili è il motivo per cui è un ottimo caso di studio per quanto riguarda le soluzioni visive impiegate. Fra gli episodi sono da notare i corti "L'erba



**Figura 2.2:** Love, death and robots - L'erba alta

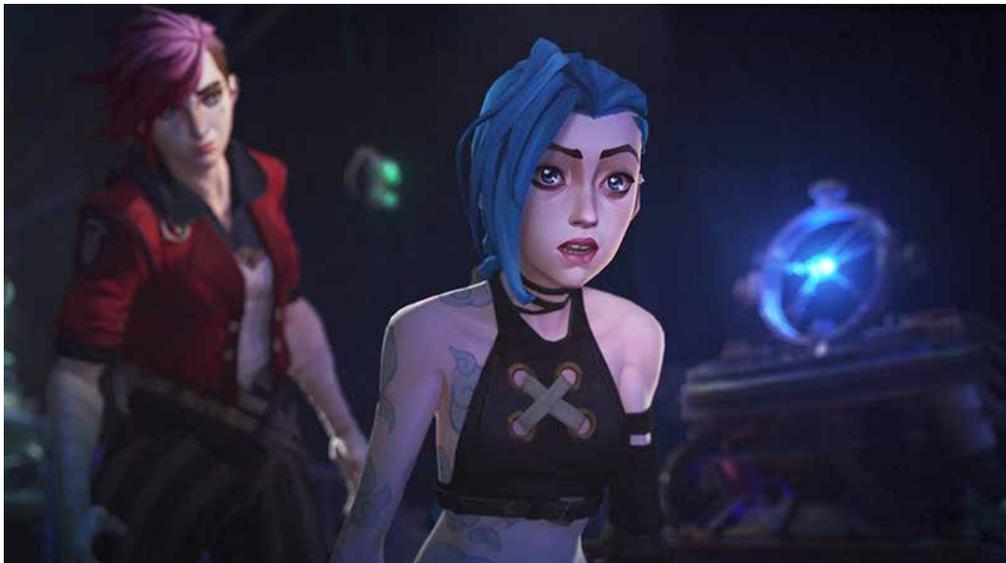
alta”(2.2) e “Il testimone” . Il primo fornisce un esempio di stile creativo con che richiama il noir, con volti espressivi e abbigliamento che si adatta all'ambientazione, oltre l'uso di ombre accentua le loro caratteristiche. "L'erba alta" fa inoltre un uso creativo del colore, usando colori saturi e le luci neon che si stagliano in modo vivido contro lo sfondo oscuro, creando un'atmosfera visivamente accattivante. Il secondo episodio in esame invece si caratterizza per l'ambientazione urbana e quasi cyberpunk. Lo stile di animazione usa un misto di animazione 3D con uso di overlay e elementi in 2D.

Anche quest'esempio ci permette di valutare le scelte artistiche impiegate ma non fornisce un modello per quanto riguarda la produzione in se. Ogni episodio è da valutare come un prodotto a parte e non fornisce un esempio pratico su una produzione seriale su più puntate, usando lo stile differente di ogni puntata il punto di forza della serie e sfruttando il senso di novità di ognuna.

## 2.3 Arcane

Arcane(2.3) è una serie televisiva d'animazione creata da Riot Games, l'azienda dietro il famoso gioco online "League of Legends". La serie è stata prodotta in collaborazione con Fortiche Productions ed è stata distribuita su Netflix. Ambientata nell'universo di "League of Legends", "Arcane" è una serie di animazione che ha debuttato nel novembre 2021.

“Arcane” rappresenta un modello non solo dal punto di vista visivo ma anche dell'ambientazione che combina elementi di cyberpunk e steampunk, creando un mondo unico che è una combinazione di tecnologia avanzata e design retrò. Dal punto di vista visivo rappresenta un riferimento sia per lo stile artistico di character design dei personaggi che dello scelte visive riguardanti lo shading e le texture. Una caratteristica di questa serie è l'uso degli ambienti realizzati in



**Figura 2.3:** Arcane

tecniche di illustrazione 2D con l'integrazione di alcuni elementi in modellazione 3D. L'integrazione fra questi è vitale per la creazione di un prodotto che deve essere coerente in ogni inquadratura. La realizzazione e l'animazione dei personaggi invece è realizzata in 3D ma a questo si aggiunge un forte uso di overlay e texture disegnate appositamente per ogni inquadratura. Infatti si può notare come cercano di mescolare il render del motore grafico con degli overlay e texture e che non interagiscono in modo coerente alla luce ma donano un aspetto visivo più vicino a tecniche tradizionali.

A livello produttivo anche questo workflow non si può applicare al nostro caso in quanto fa uso massiccio di artisti 2D che integrano le animazioni in 3D. Nel

nostro caso possiamo solo in parte aggiungere dettaglio ma sicuramente gran parte del lavoro deve essere realizzato tramite shader procedurali che facciano gran parte del lavoro.

## Capitolo 3

# Tecniche e strumenti

Dopo aver ricercato quali fossero le soluzioni artistiche che più potessero adeguarsi alla nostra soluzione produttiva si è dovuto definire gli strumenti che avremmo usato per realizzare il progetto. Fin da subito, data la precedente esperienza, si è deciso di usare come software di sviluppo sia per la parte di modellazione degli ambienti che di shading e texturing Blender.

### 3.1 Blender

Blender([3]) è un software di grafica 3D gratis e open-source con numerose funzionalità, progettato per permettere di realizzare un intero progetto di grafica 3D: modeling, rendering, animazione, rigging, video Editing, VFX, tompositing, texturing e simulazioni. Una caratteristica di blender è quella di essere un'applicazione cross-platform che può girare su Windows, macOS e Linux, grazie all'OpenGL GUI(Graphical user interface). Rispetto ai suoi competitor Blender necessita di relativa poca memoria e requisiti di driver meno stringenti.

Blender nasce all'interno dello studio di animazione olandese NeoGeo([4]). Venne creato dalla mente di Ton Roosendaal(co-fondatrice di NeoGeo) all'inizio come nuovo software interno all'azienda ma fu poi ripensato come un'applicazione che rispondesse alle necessità di artisti al fuori di questa. Fu creata quindi Not a Number(Nan) per permettere lo sviluppo e commercializzazione di Blender. In un settore della computer graphics che nel 1998 era dominato applicazioni commerciali con costi proibitivi per un utente medio, l'idea di Roosendaal era quella di di creare e distribuire gratuitamente un'applicazione 3D compatta multiplatforma. Gli investitori decisero però chiudere tutte le operazioni a causa delle vendite deludenti. Roosendaal allora decise di fondare un organizzazione senza scopo di lucro, Blender Foundation, nel 2002 per mantenere in vita il progetto. Blender

fu rilasciato sotto i termini di condizione GNU GPL che permette all'utente di di usare, copiare, modificare, ridistribuire, senza limiti.

La natura open source di Blender permette ne ha garantito il successo, permettendo a tanti artisti e studi di produzione indipendenti di realizzare progetti di grafica 3D. La comunità Blender è nota per la sua apertura e disponibilità nel condividere conoscenze e risorse, che ha portato alla creazione di numerosi tutorial, add-on personalizzati per aiutare gli utenti a imparare e utilizzare Blender al massimo delle sue potenzialità.

Con la crescente popolarità del software e di una community di sviluppatori Blender è oggi uno strumento che ha aggiunto tante nuove funzionalità che permettono di creare progetti d'animazione a livello d'industria cinematografica. Blender è stato usato in varie vesti in film come "Loving Vincent", "Next Gen" e "I lost my body".

## 3.2 Motore di rendering

Un motore di rendering è un componente immagini a partire o da un modello tridimensionale. Questo processo di rendering coinvolge il calcolo delle interazioni tra la luce, i materiali, le ombre e gli oggetti in una scena, producendo un'immagine finale o un frame animato. Essi influiscono notevolmente sulla qualità e sulla resa visiva delle immagini o delle animazioni prodotte. I motori di rendering possono variare in base alla tecnologia e agli algoritmi utilizzati per generare le immagini. I principali due tipi di motori di rendering sono:

- Path e ray tracing: simulano il percorso dei raggi di luce attraverso una scena, calcolando riflessioni, rifrazioni, ombre e altre proprietà ottiche per ottenere risultati fotorealistici.
- Rasterizer: utilizzano una tecnica di rendering basata sul pixel che è ottimizzata per la velocità e l'interattività in tempo reale.

Cycles ed Eevee([5]) sono due motori grafici integrati nel software di modellazione e rendering 3D Blender. Entrambi hanno scopi e caratteristiche differenti e sono progettati per soddisfare diverse esigenze nel processo di creazione di immagini e animazioni.

Cycles è un motore di rendering fotorealistico che si basa su un approccio di path tracing (tracciamento dei raggi). Questo significa che calcola la luce e le ombre in modo molto accurato, offrendo risultati altamente realistici. È ideale per la produzione di immagini e animazioni che richiedono una resa di alta qualità, come scene architettoniche, o rendering di personaggi per film o videogiochi. Offre funzionalità avanzate come l'illuminazione globale, il caustico, il depth of field e

l'ottica sferica. Tuttavia, poiché calcola ogni raggio individualmente, può richiedere tempi di rendering più lunghi rispetto a Eevee.

Eevee è un motore di rendering in tempo reale, progettato per fornire una visualizzazione rapida e interattiva all'interno dell'ambiente di Blender. Si basa su una tecnologia di rasterizzazione simile a quella dei videogiochi. È ideale per la produzione di anteprime, prototipi o per ottenere una resa veloce durante il processo di sviluppo, consentendo agli artisti di vedere gli effetti delle modifiche in tempo reale. Anche se Eevee non offre la stessa qualità fotorealistica di Cycles, è molto efficace per scene con una buona illuminazione in tempo reale, giochi, animazioni e video con l'obiettivo di accelerare i tempi di produzione.

La natura di questi due motori permette agli artisti 3D di utilizzare sia Cycles che Eevee in Blender. Si può utilizzare Eevee per una visualizzazione interattiva e per ottimizzare il flusso di lavoro durante la fase di progettazione e poi passare a Cycles per il rendering finale quando è necessaria una qualità ottimale. L'uso di questi due motori all'interno di Blender offre molta flessibilità nella produzione di contenuti 3D, consentendo agli artisti di bilanciare la qualità e la velocità in base alle loro esigenze specifiche.

Dopo aver analizzato entrambi i motori di rendering si è deciso di puntare su Eevee due motivazioni. La prima motivazione è dovuta al fatto che il precedente shader che è stato sviluppato è stato pensato e progettato tenendo in considerazione le funzionalità di Eevee, che fu lanciato con la versione 2.80. In particolare l'introduzione del nodo Shader to RGB (che allora era disponibile solo in Eevee) ha permesso di rompere la catena di rendering prendendo i valori RGB in uscita da un nodo di tipo shader e rielaborandoli per creare un nuovo shader.

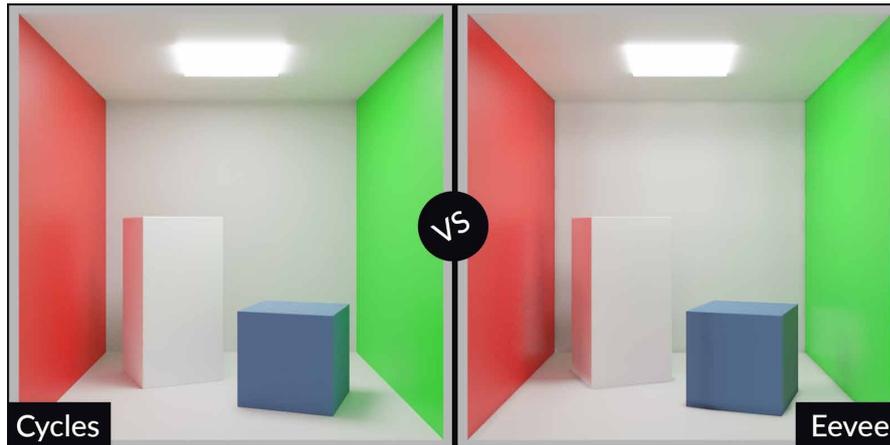
### **3.3 Differenze fra Cycles e Eevee**

Come già detto, Cycles e Eevee sfruttano due tecnologie di rendering diverse, una basata sul path tracing e una su una tecnologia rasterizer basata su OpenGL. Di seguito analizzo le principali componenti che vengono calcolate per realizzare un render da parte di Eevee e le varie differenze con Cycles.

#### **3.3.1 Illuminazione globale**

L'illuminazione globale o indiretta è una delle grandi differenze fra il motore di rendering Eevee e Cycles perché dipende in modo incisivo con il metodo di calcolo usato per il render. Nel caso di Cycles l'impatto dell'illuminazione indiretta viene calcolata in base alla traiettoria, e quindi alle informazioni contenute, del raggio di luce lungo la scena. I raggi raccolgono dati sulla componente di colore assorbita e la quantità di energia assorbita e trasmessa ogni volta che si verifica un "rimbalzo" su un oggetto. In Eevee, per sopperire alla mancanza di strumento diretti per la

misurazione di questa componente, vengono usati i light probes. Questo strumento serve per vedere la scena da più punti di vista e vedere quindi l'interazione della luce con più angolazioni e permettono di calcolare un'approssimazione dell'effetto. La natura di questo metodo implica che la componente indiretta debba essere calcolata prima in modo statico e quindi non si comporterà in modo eccellente in scenari di animazione. Una conseguenza di non usare il path tracing è quello di avere delle ombre più nette perché non vengono attenuate dalla luce che rimbalza da altre direzioni.



**Figura 3.1:** Illuminazione globale - Cycles e Eevee (CGCOOKIE.COM)

### 3.3.2 Riflessioni

Anche in questo caso le differenze fra un metodo e l'altro sono visibili e dipendono dal modo fondamentale con cui lavorano i due motori di render. Cycles la componente di riflessione viene calcolata in modo intuitivo in base alla traiettoria e la distribuzione dell'energia luminosa nella scena.

Al contrario Eevee sfrutta il metodo dello screen space reflection che calcola il riflesso in base a delle trasformazioni geometriche dell'immagine riflessa, che viene ruotata e adattata in base alla direzione della riflessione sull'oggetto riflettente. Anche in questo caso è possibile utilizzare dei light probes per sopperire a delle mancanze del modello, come il fatto che non funziona su oggetti che non sono visibili alla camera di render. In modo simile a quello che fa nel caso dell'illuminazione diretta, sfrutta dei piani da diversi punti di vista per avere più informazioni sugli oggetti in questione e fornire un risultato più accurato in fase di rasterizzazione. Si può effettuare un bake delle riflessioni (necessario per light probes che non sono planari) per avere una velocità di calcolo maggiore ma non è compatibile con delle animazioni.

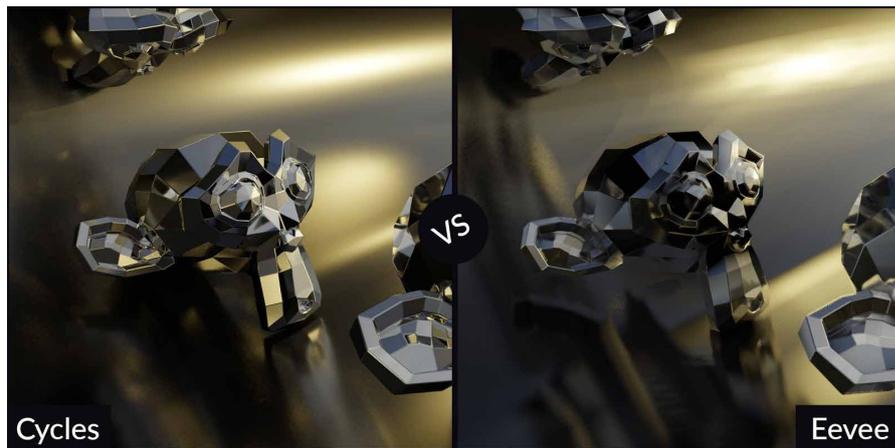


Figura 3.2: Riflessione - Cycles e Eevee (CGCOOKIE.COM)

### 3.3.3 Rifrazioni

In Cycles la rifrazione viene calcolata in base a come la traiettoria della luce si comporta lungo il percorso dentro l'oggetto e quindi simulando come la luce si comporta nella realtà. Un oggetto che ha una trasmittività diversa da un altro comporterà una diversa traiettoria del raggio luminoso. Nel caso di Eevee la componente dell'immagine dietro (dal punto di vista della camera) a un oggetto trasmissivo viene calcolata attraverso delle trasformazioni delle normali delle mesh occluse che ha come risultato quello di distorcere l'oggetto rifratto.



Figura 3.3: Rifrazioni - Cycles e Eevee (CGCOOKIE.COM)

### 3.3.4 Trasparenza

Con Cycles la trasparenza è calcolata in modo che quando un raggio incontra una superficie con un materiale trasparente, venga calcolata la rifrazione del raggio in base all'indice di rifrazione del materiale. L'indice di rifrazione determina quanto il raggio viene piegato mentre attraversa il materiale. In Eevee la trasparenza di un oggetto viene data dalla blend mode che definisce come il colore di una superficie viene aggiunto al buffer dei colori dopo il calcolo. Ci sono diverse modalità tra cui Opaque, che sovrascrive il colore precedente, Alpha Clip, che sovrascrive solo se il valore alpha è sopra una soglia, Alpha Hashed, che usa una soglia casuale e Alpha Blending, che sovrappone il colore dal più vicino al più lontano dal punto di vista della camera.

### 3.3.5 Ambient occlusion

L'ambient occlusion (AO) è un effetto visivo utilizzato nei motori di rendering per simulare l'ombreggiatura aggiuntiva nelle aree di una scena dove gli oggetti sono vicini l'uno all'altro o dove la luce ha difficoltà a raggiungere. In altre parole, l'ambient occlusion aggiunge ombre fittizie per enfatizzare le cavità e le zone in cui gli oggetti sono più ravvicinati. Nel caso di Cycles coinvolge il tracciamento dei raggi per determinare l'occlusione in base alla distanza nello spazio 3D fra le superfici. In Eevee si usa la Screen Space Ambient Occlusion che calcola l'occlusione in base alla profondità di ogni pixel rispetto alla telecamera (depth map). Viene infatti calcolata la distanza fra le superfici della scena renderizzata in base alla loro posizione contenuta nella mappa di profondità dal punto di vista della camera.

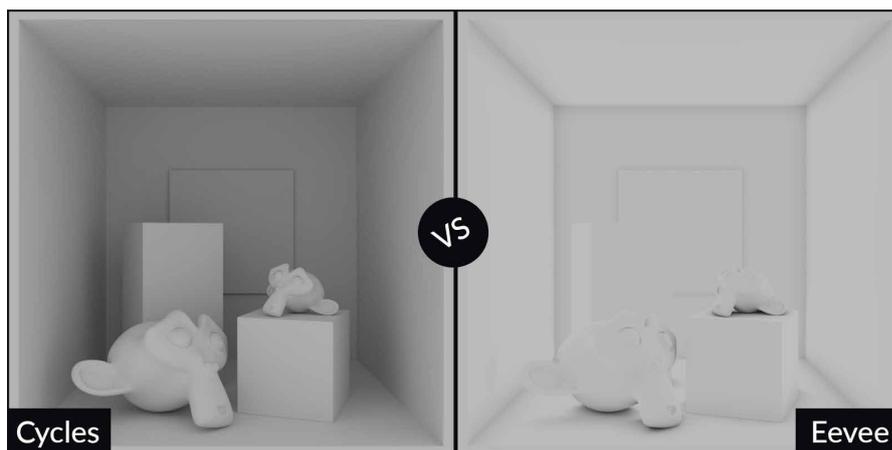


Figura 3.4: Ambient occlusion - Cycles e Eevee (CGCOOKIE.COM)

### 3.3.6 Ombre

La tecnica di calcolo delle ombre in Cycles è intuitiva in base al metodo utilizzato, ovvero viene calcolato l'impatto dei raggi luminosi in una data zona e calcolando la distribuzione dell'energia luminosa. In Eevee il metodo per calcolare le ombre consiste nello sfruttare lo shadow mapping. Questa tecnica ha l'obbiettivo di calcolare l'impatto delle ombre ponendosi dai punti di vista di ogni sorgente di luce nella scena e comparando i risultati di ogni immagine calcolata per capire i punti in ombra. Un'altra tecnica per rendere le ombre più diffuse è impostando nelle luci il parametro diffuse che simulerà un'ombra più morbida inserendo più luci vicine al punto di origine della sorgente luminosa e calcolando l'impatto di ognuna per fornire ombre meno nette. Nelle impostazioni di render inoltre è possibile lavorare sui parametri soft shadows e contact shadows per fornire un risultato migliore ma con maggior onere di calcolo.

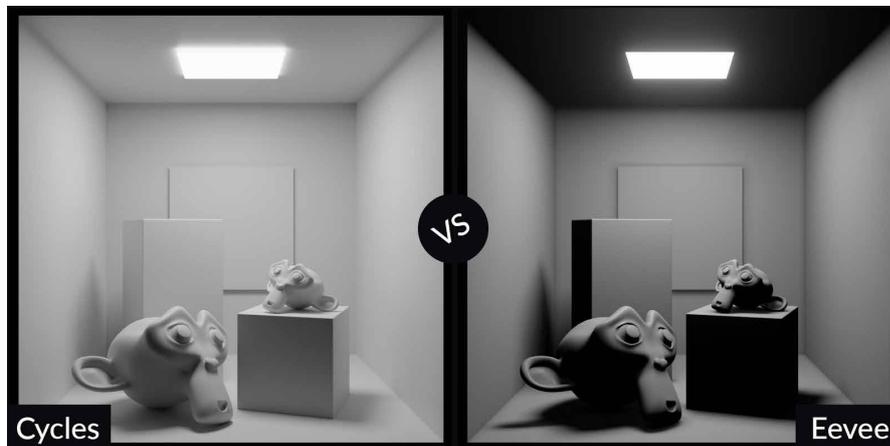
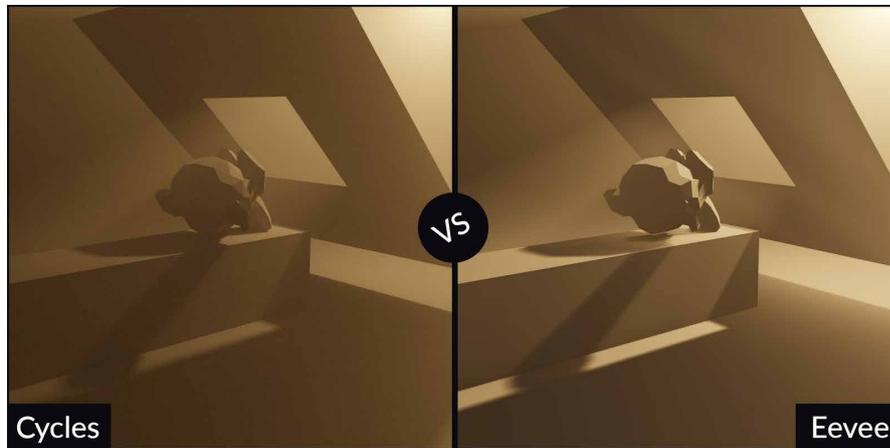


Figura 3.5: Ombre - Cycles e Eevee (CGCOOKIE.COM)

### 3.3.7 Luce volumetrica

Eevee consente di aggiungere effetti volumetrici alle scene in modo relativamente veloce e interattivo, anche se in modo approssimato rispetto a tecniche più avanzate di tracciamento dei raggi come quelle utilizzate in Cycles. Per luce volumetrica si intende la possibilità di visualizzare fasci di luce nella scena 3D. I risultati fra i due motori in questo caso sono abbastanza comparabili anche se utilizzano tecniche differenti. Cycles usa sempre il principio del path tracing in congiunzione alla depth map e la shadow map per determinare la componente volumetrica della luce. Eevee utilizza una tecnica basata sullo screen space per approssimare l'interazione della luce con i volumi nella scena. Ciò significa che l'illuminazione volumetrica è

calcolata in base a ciò che è visibile sullo schermo, anziché tracciare raggi attraverso l'intero spazio tridimensionale.



**Figura 3.6:** Luce volumetrica - Cycles e Eevee (CGCOOKIE.COM)

### 3.3.8 Subsurface scattering

Il subsurface scattering (SSS) è un effetto visivo che modella il comportamento della luce quando attraversa e interagisce con materiali semi-trasparenti o traslucidi, come la pelle umana, la cera o il latte. In Blender, il subsurface scattering è implementato per creare materiali più realistici e simili ai materiali organici. Quando la luce colpisce una superficie con subsurface scattering, una parte della luce penetra nel materiale e rimbalza più volte, invece di riflettere direttamente. All'interno del materiale, la luce può subire diffusione e dispersione, dando luogo a un effetto di "radiazione" di luce attraverso il materiale. Questo fenomeno è particolarmente evidente in oggetti traslucidi, dove la luce può penetrare e uscire da diverse parti del materiale. Eevee si basa su profili pre-calcolati di scattering per materiali diversi. Questi profili descrivono come la luce si diffonde all'interno di un materiale in base alle sue proprietà e vengono successivamente memorizzati nelle texture. Durante il rendering, Eevee utilizza le informazioni basate sullo screen space per approssimare l'effetto di scattering.



**Figura 3.7:** Subsurface scattering - Cycles e Eevee (CGCOOKIE.COM)

### 3.3.9 Profondità di campo

La differenza di risultati riguardo la profondità di campo fra Cycles e Eevee sono marginali. Seppure Cycles calcoli questo parametro in base al reale comportamento fisico dell'interazione delle luci e la focale della lente, Eevee ottiene dei risultati simili sfruttando delle approssimazioni in post processing dell'immagine. Sfruttando un depth buffer per le informazioni di distanza e le impostazioni di lunghezza focale e apertura calcola le parti dell'immagine a cui applicare un algoritmo di blur. Le uniche differenze fra i due motori di render si possono riscontrare nelle transizioni tra aree a fuoco o meno.



**Figura 3.8:** Profondità di campo - Cycles e Eevee (CGCOOKIE.COM)

### 3.3.10 Motion blur - Eevee

Cycles calcola il motion blur sempre attraverso il calcolo del percorso dei raggi nella scena e le impostazioni della lente come la shutter curve che determina il comportamento dello shutter virtuale. Al contrario, Eevee genera vettori di movimento per ogni pixel della scena. Questi vettori indicano la direzione e la velocità del movimento di ogni punto nell'immagine durante ogni frame dell'animazione. Dopo il rendering di ogni frame, Eevee applica l'effetto di motion blur come passo di post-processing([6]). Utilizzando i vettori di movimento generati in precedenza, l'immagine viene sfocata in modo proporzionale al movimento apparente di ogni pixel. Questa tecnica può produrre artefatti lungo i bordi se il movimento è troppo complesso. Oltre a questo processo Eevee usa la tecnica di Accumulation Motion Blur, che consiste nel dividere il render in più step temporali e accumulare il risultato. Ad ogni step corrisponde ad una completa rivalutazione della scena calcolata in post processing e offre buoni risultati per un numero di step elevati.

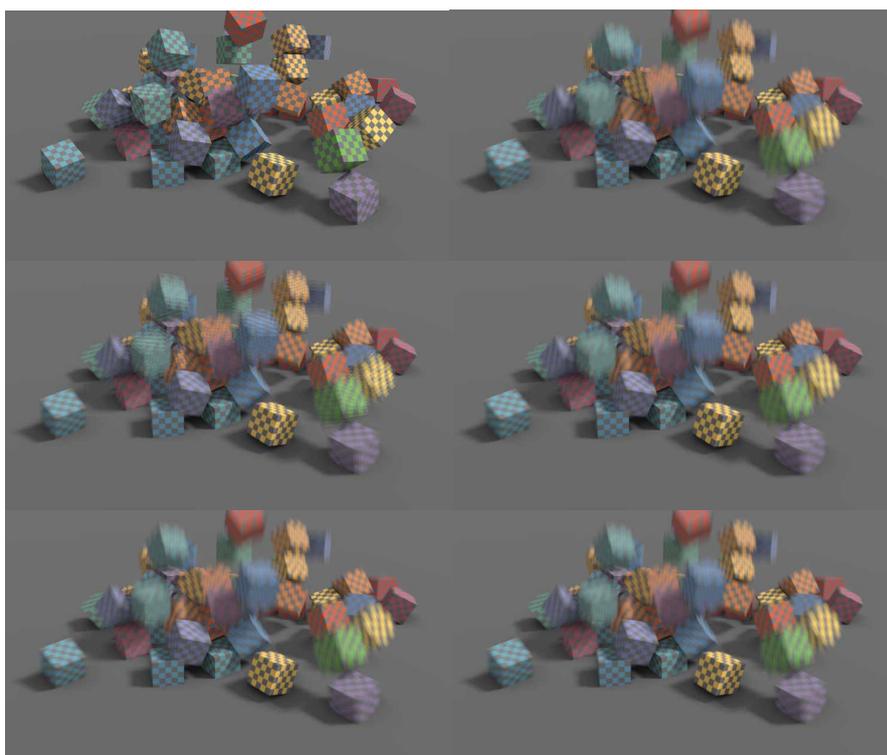


Figura 3.9: Motion blur([6])

### 3.3.11 Bloom

L'effetto di bloom è presente esclusivamente nel motore di render Eevee. Il bloom migliora il rendering delle aree luminose nella scena, creando un effetto di glow luminoso attorno alle sorgenti luminose ad alta intensità. Questo effetto simula il modo in cui le fotocamere del mondo reale catturano e mostrano la luce intensa. In prima analisi, Eevee identifica i pixel luminosi nell'immagine, ovvero i pixel che superano una certa soglia di luminanza. I pixel luminosi vengono estratti e isolati dal resto dell'immagine a cui viene applicato un effetto di blur. Questo passaggio crea un alone o un bagliore attorno alle aree luminose, simulando l'effetto della dispersione intensa della luce o del riversamento nei pixel adiacenti. Eevee fornisce parametri regolabili per controllare l'intensità, le dimensioni e il colore dell'effetto di bloom.



**Figura 3.10:** Bloom

# Capitolo 4

## Produzione

In questo capitolo illustro i vari passaggi che abbiamo svolto durante il processo di preproduzione e produzione. Partendo dai materiali e documenti che avevamo a disposizione è stato necessario effettuare le varie fasi di preproduzione per realizzare la scena del nightclub dell'episodio pilota. Finita questa fase preliminare si è entrati nella fase di realizzazione degli asset dividendo il lavoro in più task che illustrerò in seguito.

### 4.1 Preproduzione

Dopo aver parlato con il team di lavoro la prima cosa che abbiamo cercato di fare è stato analizzare il materiale di preproduzione già presente. Come già detto la serie è entrata in preproduzione nel 2017 e si è susseguita in più fasi. Il materiale a disposizione era :

- Documento di concept della serie
- Sceneggiatura dell'episodio pilota
- Character sheet e sketch dei personaggi principali(4.1)
- Sketch e concept art del night club(4.2)
- Reference del nightclub

I character sheet dei personaggi principali è stato essenziale per la modellazione dei personaggi della sceneggiatura in esame. Il documento contiene informazioni sull'aspetto, personalità e background del personaggio in questione ed è riferimento per capire le azioni dei personaggi all'interno della scena. Gli sketch sono stati altrettanto importanti per la modellazione dei personaggi, fornendo punti di vista diversi per la loro realizzazione. La scena dell'episodio che ci è stato chiesto di

## Nadya

Nome Completo: Nadya Sinkamen



**Età:** 19  
**Sesso:** Female  
**Altezza:** 172 cm **Peso:** 54kg  
**Carnagione:** Pallida  
**Capelli:** biondi, media lunghezza, leggermente mossi. Rasati sul lato sinistro  
**Occhi:** Blu  
**Palette abiti:** Blu (#682b4f) / Lilla  
**Palette dettagli:** Azzurro (c7d9e8) / Grigio

### ASPETTO

Fisico slanciato e allenato. Presenta un impianto biomeccanico dietro il collo, tramite il quale si inietta la dose giornaliera di farmaco per la gestione di sinestesia ed iperestesia. Occhi blu. Corti capelli biondi, rasati sul lato sinistro del cranio. Le punte sono tinte di blu. Alcuni piercings all'orecchio sinistro, anello alla narice destra. Grosso insetto ("Alep", vedi scheda) che l'accompagna ovunque e le sosta spesso sulla spalla sinistra.

### PERSONALITÀ

**OUTSIDE** - Curiosa, tenace, inquisitiva, sarcastica. Ostenta cinico distacco.

**INSIDE** - Insicura, fortemente legata ai suoi amici, spaventata dal proprio male

### BACKGROUND

Figlia di due dipendenti della Pharmacopia, residenti nella seconda città più grande del paese, Nadya s'è trasferita nella capitale per frequentare l'università e divenire entomologa comportamentale. In una società in cui ogni persona soffre di malattie o menomazioni genetiche, Nadya tiene sotto controllo il disturbo che l'accompagna fin dalla nascita con un impianto che le fornisce la sua dose giornaliera di medicinali. Nadia infatti soffre di sinestesia ed iperestesia, che le causano percezioni amplificate e confuse al punto d'indurle (a volte) vere e proprie allucinazioni psichedeliche.

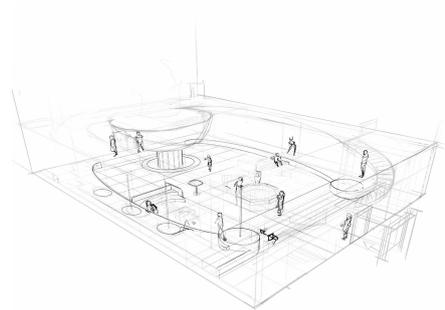


Figura 4.1: Character sheet

Figura 4.2: Sketch night club

Figura 4.3: Documenti preproduzione

sviluppare è ambientata in un night club all'interno della città. Il night club è un posto frequentato da tutti gli ambienti sociali e in cui ed è anche un luogo in cui di nascosto si tengono delle battaglie di insetti(creature tipiche del mondo di Reverie Dawnfall). Gli sketch già presenti riguardo questo ambiente sono stati una guida chiara per la ricerca di reference e modellazione della scena. Per prima cosa abbiamo cercato delle reference che sin da subito potessero dare un'idea del tipo di contesto e palette di colori. Queste reference sono state una risorsa che ci ha guidato durante tutta la modellazione della scena e degli oggetti necessari, nonché nello spoglio della sceneggiatura.

## 4.2 Spoglio sceneggiatura

Il passo successivo dopo aver preso in analisi i documenti relativi ai personaggi e l'ambiente è stato quello di fare uno spoglio della sceneggiatura. Lo spoglio della sceneggiatura consiste nel leggere la sceneggiatura e individuare tutti gli elementi che saranno necessari per la realizzazione del prodotto visivo. Durante lo spoglio abbiamo individuato gli elementi che dovevano rientrare all'interno della scena e li

abbiamo divisi in :

- Personaggi
- Animali
- Props
- Oggetti di scena
- Sound effects
- Lighting

La fase di spoglio della sceneggiatura è stata fondamentale per capire oltre tutti gli asset necessari, la mole di lavoro e come distribuirlo all'interno del nostro team. I task che abbiamo individuato sono stati divisi in object modelling, character modelling, cloth modelling, shading, lighting, texturing e animation. Io mi sono incaricato della parte di shading, texturing e lighting di cui parleremo nel corso di questa tesi.

Aa Name	Tags	Status	Person	Modello	Rigging	Texture
Abiti breather	Cloth	In progr...	Chiara Sanfil	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Abiti Grivil	Cloth	In progr...	Chiara Sanfil	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Abiti Daimon	Cloth	Not start...	Chiara Sanfil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Abiti Guardie	Cloth	Not start...	Chiara Sanfil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Granata fumogena	Props	Not start...	Chiara Sanfil	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Lecca lecca	Props	Not start...	Chiara Sanfil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Apparecchietto(vicino la porta)	Props	Not start...	A Andrea Loro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
piccolo congegno dall'aspetto raffazzonato	Props	Not start...	A Andrea Loro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Suono di un vecchio modem 56Kpbs	Sound FX	Not start...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ping	Sound FX	Not start...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gatto cyborg	animali	Not start...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Struttura sala	Enviroment	Not start...	E Eugenio Nic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tavolo piccolo a forma di arena	Enviroment	Done	A Andrea Loro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
bancone bar	Enviroment	Done	A Andrea Loro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
bottiglie bar	Enviroment	Done	A Andrea Loro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
serratura	Enviroment	Not start...	A Andrea Loro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
porta della gambling den	Enviroment	Done	A Andrea Loro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
bicchieri	Enviroment	Done	A Andrea Loro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tavoli	Enviroment	Done	A Andrea Loro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 4.4: Breakdown spoglio sceneggiatura

## 4.3 Storyboard

Lo storyboard è una sequenza di disegni, illustrazioni o immagini che rappresentano visualmente la successione delle scene di un film, di un video, di un'animazione o di qualsiasi altro tipo di produzione visiva. Ogni immagine nello storyboard rappresenta una singola "inquadratura" aiutando a comunicare la struttura narrativa, la composizione visiva e la sequenza temporale del prodotto. Lo storyboard può essere realizzato in modo tradizionale con disegni a mano su carta o in modo digitale utilizzando software specifici. È uno strumento flessibile che può essere utilizzato in vari campi, tra cui cinema, animazione, pubblicità, fumetti e altro ancora.

Lo storyboard è un documento essenziale in fase di produzione perché comunica in modo efficace e sintetico molte informazioni che in fase di sceneggiatura mancano, come movimenti di camera e dei personaggi. In genere in uno storyboard viene indicato:

- I movimenti della macchina da presa
- Il tipo di focale
- Tipo d'inquadratura
- Descrizione della scena
- Location
- Scena
- Location
- Luce



private. I nostri protagonisti sono in questo locale perché devono entrare in una stanza segreta dove si tengono lotte clandestine fra insetti mutanti e recuperare l'insetto di Nadya. La scena come descritta da sceneggiatura comporta l'uso di molti establishing shot (delle inquadrature che stabiliscono il contesto di una scena), molte inquadrature di dialogo e alcune inquadrature più movimentate d'azione.

Per prima cosa abbiamo proceduto definendo in maniera molto approssimativa le figure e i movimenti. In un secondo momento abbiamo realizzato un documento con delle illustrazioni più evocative delle inquadrature e le varie descrizioni necessarie alla produzione.

## 4.4 Animatic

Dopo lo storyboard è prassi comune realizzare un animatic, un video che contiene le inquadrature dello storyboard montate in sequenza. Rispetto allo storyboard l'animatic contiene ulteriori informazioni :

- Quali e quante inquadrature sono necessarie
- Il timing delle inquadrature, indicando quanto permane sullo schermo e in che momento. Questo aiuta a dare un'idea del ritmo e della durata complessiva del progetto
- Azioni che accadono nell'inquadratura, tramite animazioni di movimenti degni di nota
- Audio di riferimento per dare un'idea più completa di come la sequenza si tradurrà in termini di narrativa audiovisiva
- Scelte di montaggio e transizioni che danno un'idea del ritmo e il flusso narrativo del progetto

Un animatic può essere realizzato tramite varie tecniche, come ad esempio l'animazione tradizionale o tramite software di modellazione 3D. Nel nostro caso abbiamo scelto di realizzare l'animatic utilizzando Blender(4.6). Ciò ci ha dato modo di capire, nello stesso software che avremmo utilizzato per l'animazione, come i personaggi si sarebbero mossi nell'ambiente. Inoltre animare su un software di animazione 3D ci ha permesso di potere ridefinire in modo veloce eventuali cambi. Posizionare le camere virtuali e definire per ciascuna il tipo di lunghezza focale è stato inoltre molto utile per capire come in ogni inquadratura venissero rappresentati personaggi in relazione all'ambiente; se si dovesse usare una focale più lunga o corta per capire la porzione di spazio che dovesse essere ripresa in camera. Le interfacce di editing e animazione di Blender sono state molto utili

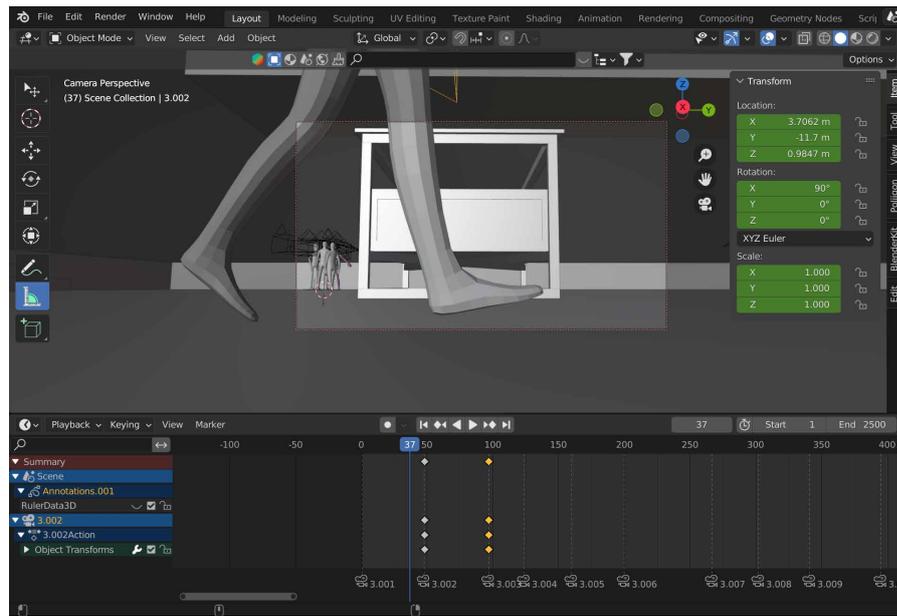


Figura 4.6: Animatic

perché forniscono degli strumenti che permettono di editare in modo veloce le varie inquadrature e le animazioni a livello temporale. Tramite la creazione di marker per ogni inquadratura è stato possibile definire in modo semplice il temporizzazione di ogni shot e cambiarlo in modo intuitivo con un drag dell'icona del marker sulla strip di editing.

Definire l'animatic su blender ci ha permesso inoltre di realizzare una scena che potesse essere una sorta di floor plan (un piano o schema dettagliato di un set o di una scena in cui sono rappresentati movimenti di camera e la disposizione spaziale degli elementi).

## 4.5 Workflow

Definire i task principali, come distribuirli nel team di lavoro e come integrarli nel processo produttivo è stata una delle sfide che abbiamo dovuto risolvere per continuare la produzione del progetto. Come scritto in precedenza i task principali per la realizzazione del progetto sono stati individuati in :

- Object modelling: la creazione e modellazione di tutti gli asset 3D che rappresentassero oggetti e quindi non persone o animali con un rig da animare.
- Character modelling: la modellazione dei corpi, volti e capelli dei personaggi compresi di rigging per effettuare in seguito l'animazione

- Cloth modelling: la modellazione dei vestiti dei personaggi e la successiva applicazione ai corrispettivi modelli
- Shading: sviluppo di vari shader per definire i materiali dei vari asset 3d nella scena e modo di interazione della luce nelle inquadrature
- Lighting: posizionamento delle luci nell'ambiente 3D per definire l'illuminazione nella scena e nelle inquadrature
- Texturing: realizzazione delle texture finali in software di illustrazione 2D, a partire da quelle realizzate con lo shader, tramite l'utilizzo di tecniche di camera projection
- Animazione: animare tramite Blender tutti i personaggi e oggetti in base ai movimenti definiti in fase di storyboard e animatic.

Per prima cosa ogni membro del team di lavoro, costituito da 5 persone, ha affrontato una fase di ricerca delle tecniche e tool migliori per la realizzazione dei vari task(4.7). In seguito un'analisi delle varie procedure di lavoro richieste per ogni task ha richiesto una definizione di workflow in modo tale da ottimizzare il processo produttivo in base a la mole di lavoro richiesto e quali potessero essere i processi realizzati parallelamente e quali in modo sequenziale. Per definire l'organizzazione del progetto abbiamo usato Notion una piattaforma di produttività che offre una varietà di strumenti per la gestione delle informazioni e la collaborazione con altri utenti. Per le prove da effettuare sullo shader il lavoro che ho fatto è stato realizzato in modo asincrono senza aspettare i vari modelli da realizzare. Per testare però gli shader in un ambiente che fosse simile a quello finale ho deciso di realizzare io il modello 3D della stanza. Ciò mi ha permesso di realizzare le varie luci e posizzionarle nell'ambiente in attesa che fossero realizzati e posizionati gli asset rimanenti. Infine dopo aver creato tutti i modelli ho potuto realizzare i vari materiali per i modelli e adattare lo shader in base al modello e la luce.

Anche la realizzazione degli oggetti è stata realizzata in modo parallelo al resto per poi posizionare gli oggetti nell'ambiente in base a quello che era stato definito in fase di preproduzione. Per i task di modellazione dei personaggi e dei vestiti invece si è dovuto strutturare il lavoro in modo tale che i modelli dei personaggi fossero disponibili in tempo affinché chi si occupasse di realizzare i vestiti(realizzati con un software differente) potesse realizzarli in base alle proporzioni del modello. L'animazione della scena è stato definito come l'ultimo processo produttivo prima del render effettivo del prodotto.

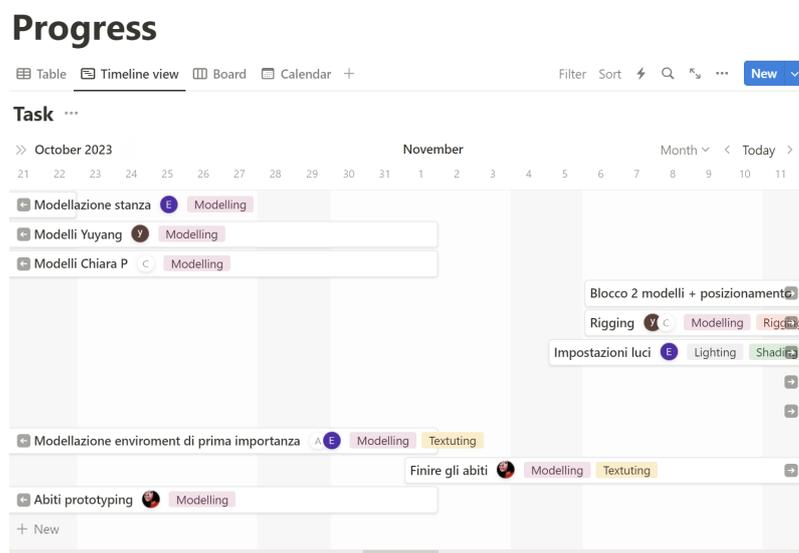


Figura 4.7: Gantt

# Capitolo 5

## Shading

Il lavoro di questa tesi si basa principalmente sulla ricerca e applicazione di uno shader che realizzi lo stile grafico individuato in fase di preproduzione. Per raggiungere il risultato voluto si è dovuto passare attraverso una fase di sperimentazione per poi applicare i vari materiali alle mesh progettate per la serie. La lavorazione di questi shader è stata effettuata, come già evidenziato, su Blender e sul motore di render Eevee, quindi introdurremo l'ambiente di sviluppo che offre la piattaforma di computer grafica per realizzare gli shader che vengono definiti materials e si userà quindi la nomenclatura del software come diffuse, emission, roughness.

Sono stati effettuati dei test su oggetto standard e poi dei test su dei personaggi campione per individuare se il look visivo era quello voluto.

### 5.1 Obiettivi

L'obiettivo dello shader da sviluppare era quello di migliorare quello precedente da alcuni punti di vista. Lo shader precedente dava buoni risultati per la gestione del colore e degli highlight e shadow ma non permetteva di gestire bene altri aspetti ed era molto macchinoso da usare.

Dopo l'analisi dei casi studi presentati nel capitolo Stato dell'arte e in base agli obiettivi che si ci era posti in fase di preproduzione, si è deciso di fissare come obiettivo il miglioramento relativamente a questi aspetti:

- Sistema di coordinate migliore
- Retino a righe
- Halftoning
- Pattern a esagoni
- Outline

## 5.2 Ambiente shading di Blender

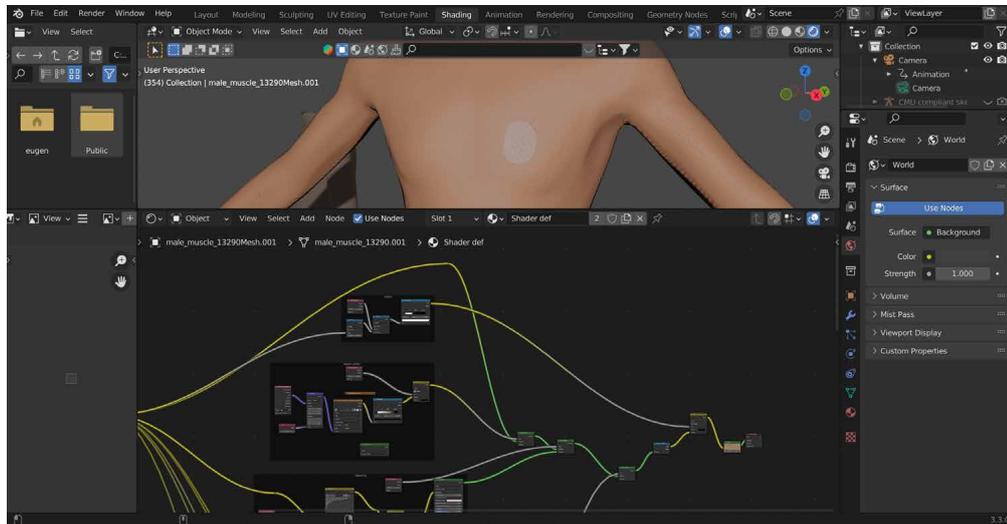


Figura 5.1: Shader editor

Lo shading a nodi in Blender è un approccio visuale e nodale alla creazione di materiali. Invece di utilizzare un'interfaccia basata su testo, come nei linguaggi di shading tradizionali, gli artisti usano il "Shader Editor" di Blender per collegare nodi, ciascuno dei quali rappresenta una funzione o una proprietà specifica del materiale. Questo metodo offre un modo intuitivo e flessibile per controllare l'aspetto dei materiali.

Blender divide i nodi in alcune categorie che hanno diverse:

- **Input:** sono i nodi che contengono le informazioni su come la luce interagisce sulla mesh dell'oggetto, dando informazioni per esempio sull'ambient occlusion, colore, diffusione e rifrazione. Inoltre contengono informazioni sulla geometria dell'oggetto in questione come il nodo di Texture coordinates che fornisce i dati vettoriali in base ai diversi sistemi di coordinate. Sono sempre usati in ingresso nel sistema di nodi perché contengono le informazioni necessarie a modificare l'apparenza dell'oggetto.
- **Output:** è la categoria di nodi che contiene AOV Output e Material output. Sono i nodi finali responsabili di modificare l'oggetto su cui viene applicato il materiale in base a agli input di ingresso che riguardano la superficie (quello più usato), volume e il displacement e sono preceduti sempre da un nodo di shader tranne nel caso si usi un displacement.
- **Shader:** in questa categoria sono presenti tutti i nodi che fungono da shader e quindi dei nodi che tramite dei modelli matematici prendono le informazioni

in ingresso che riguardano l'interazione della luce e forniscono un output che fornisce l'effetto desiderato di quello specifico shader. Il più usato è il Principled BSDF che contiene molti input d'ingresso e punta ad un output completo e fotorealistico (anche se non necessariamente), mentre altri shader meno complessi forniscono risultati per particolari effetti della luce come i Diffuse ed Emission. Gli output degli shader possono essere a loro volta mixati insieme tramite dei nodi Mix shader e un fattore di mix, che utilizza informazioni scalari o un'immagine in scala di grigi come valori per controllare il mix tra due elementi differenti.

- **Texture:** sono i nodi responsabili di importare delle immagini da usare come texture per i modelli, come Image texture, ma anche di creare delle texture procedurali, ovvero create tramite dei calcoli a partire da valori d'ingresso. Di quest'ultima categoria fanno parte i Noise texture e Gradient texture.
- **Color:** questi nodi manipolano le informazioni di colore permettendo di trattare i dati come dei vettori RGB e quindi effettuare trasformazioni vettoriali e trattando il colore in modo lineare. Sono utili per effettuare operazioni di trasformazione del colore classici, come operare dalla saturazione, tono e intensità o agire sulla gamma dinamica. Tramite il nodo Mix è possibile mixare più canali tramite un fattore di mix e con diverse modalità di blend, come per esempio Overlay, lighten, multiply, darken.
- **Vector:** sono un insieme di nodi utilizzati per manipolare e gestire le informazioni vettoriali. Lavorano con vettori che rappresentano direzioni, spazi, informazioni geometriche e sistemi di coordinate dell'oggetto su cui è applicato il materiale. I nodi Bump e Normal map sono importanti per creare l'informazione vettoriale a partire da immagini 2D mentre il vettore di Mapping è spesso usato subito dopo il nodo di Texture coordinates per manipolare l'informazione data.
- **Converter:** sono un insieme di nodi utilizzati per convertire, manipolare e gestire i dati di input e output nei materiali. Questi nodi sono utili per adattare i dati in modo che possano essere utilizzati in diversi contesti all'interno dello shader e convertendo un tipo di dato in ingresso, come quello del float scalare, in un altro tipo d'informazione, come quella di colore. Sono molto utili i nodi di tipo math per effettuare operazioni matematiche fra valori scalari e ColorRamp che consente di controllare la distribuzione dell'output colore lungo una scala di valori.
- **Nodi organizzativi:** sono i nodi di tipo Layout e Group che permettono di organizzare visualmente i nodi all'interno dell'editor di shading.

## 5.3 Shading procedurale

Il metodo procedurale è un approccio alla generazione di contenuti, spesso utilizzato nel computer graphic, che si basa su regole e algoritmi piuttosto che su risorse esterne come immagini o modelli. Questo metodo consente la creazione di contenuti in modo automatico e parametrico, offrendo una maggiore flessibilità e adattabilità. Questo tipo di approccio è fondamentale per la realizzazione di shader che possano essere facili da usare e adattabili ai diversi modelli e contesti, non potendo contare sull'apporto del lavoro di un illustratore 2D.

Nel metodo procedurale, vengono definite regole e algoritmi per generare dati o contenuti. Ad esempio, invece di utilizzare un'immagine di erba per un terreno in un ambiente 3D, si potrebbe utilizzare un algoritmo procedurale che genera erba in base a parametri come altezza, densità e colore. La proceduralità si basa spesso sulla parametrizzazione, il che significa che è possibile regolare i risultati modificando i parametri delle regole o degli algoritmi. Questo offre un elevato grado di flessibilità e facilita l'iterazione e la personalizzazione del contenuto generato. Si desidera ottenere una varietà di risultati o dettagli senza dover creare manualmente ciascun elemento. Poiché i contenuti sono generati al volo tramite algoritmi, il metodo procedurale può essere meno oneroso in termini di risorse rispetto all'utilizzo di risorse esterne, come immagini o modelli 3D.

L'ambiente di Blender che è basato sui nodi è congeniale per questo tipo di approccio, dato che è facile visualizzare parametri, regole e relazioni che producono il risultato finale.

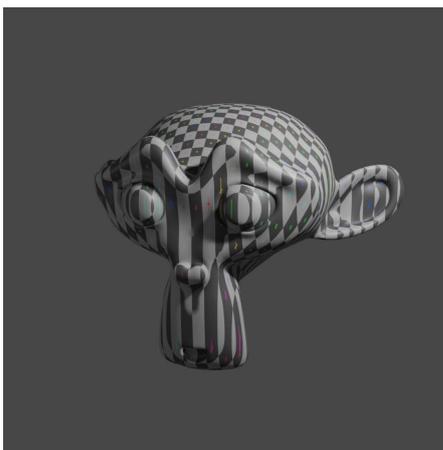
## 5.4 Sistemi di coordinate

Uno dei parametri che più influenza il risultato di uno shader è quello del sistema di coordinate scelto. Scegliere un sistema di coordinate influenza come le informazioni sulla geometria dell'oggetto vengono rilevate e quindi influenza il risultato finale, soprattutto nella fase di animazione. Il nodo più importante da questo punto di vista è quello del Texture coordinate che fornisce sette modi diversi di rappresentare le coordinate nello spazio di un oggetto ma prenderemo in analisi anche i sistemi di coordinate del nodo Camera data, Tangent e Object Info.

### 5.4.1 Generated

Quando si utilizzano le coordinate di tipo generated, Blender genera automaticamente le coordinate di texture in base alle dimensioni e alla forma dell'oggetto. Questo significa che le coordinate sono generate proceduralmente e dipendono dalla geometria dell'oggetto stesso. Le coordinate texture di tipo generated forniscono

un sistema di mappatura 3D, il che significa che ogni punto sulla superficie dell'oggetto ha tre coordinate: X, Y e Z. Queste coordinate possono essere utilizzate per determinare come la texture viene avvolta intorno all'oggetto. Le coordinate vengono calcolate a partire da una bounding solidale all'oggetto, ponendo un valore da 0 a 1 in base allo spazio che occupano del cubo. Cò implica che più un oggetto è simmetrico meno distorsioni saranno presenti.



**Figura 5.2:** Coordinate generated

## 5.4.2 Normal

Si riferiscono a un sistema di coordinate basato sulle normali della superficie di un oggetto. La normale di un punto su una superficie è una linea che è perpendicolare a quella superficie in quel punto. Ogni punto sulla superficie di un oggetto ha una normale associata, che indica la direzione perpendicolare alla superficie in quel punto. Le coordinate normal (possono essere)sono normalizzate, cioè il vettore della normale viene reso unitario, per permettere di utilizzare le coordinate per calcoli come riflessioni o ombre.



**Figura 5.3:** Coordinate normal

### 5.4.3 UV

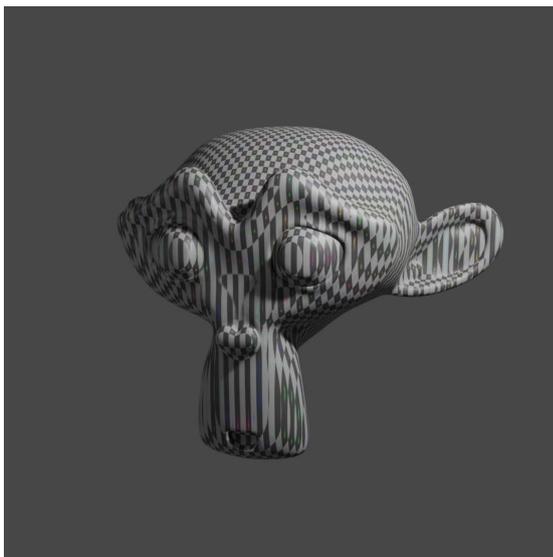
Le coordinate UV si riferiscono a un sistema di coordinate bidimensionale utilizzato per mappare texture su oggetti 3D. Questo sistema è usato per definire come le immagini o le texture vengano proiettate sulla superficie di un oggetto in modo intuitivo e controllato in base all'UV unwrap della mesh creato. L'UV unwrap è il processo tramite il cui la superficie di un oggetto viene appiattita in modo tale da poter applicare un'immagine su di essa. L'UV mapping coinvolge la proiezione delle coordinate UV sulla superficie dell'oggetto che ha subito l'UV unwrap, in modo che corrispondano ai dettagli della texture. Questo metodo è molto utile se bisogna associare dettagli particolari a punti della mesh.



**Figura 5.4:** Coordinate UV

#### 5.4.4 Object

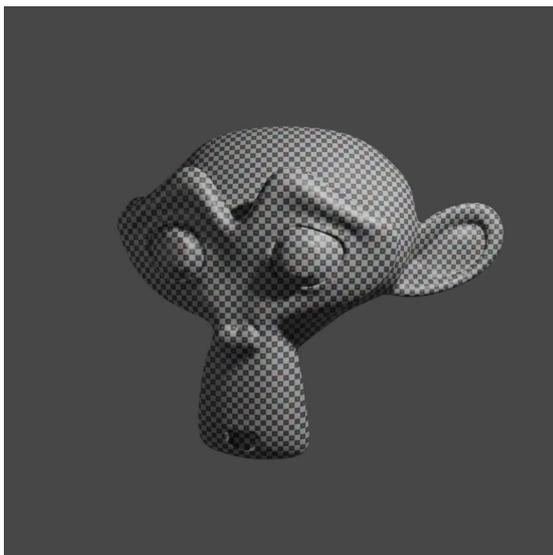
Quello object è un sistema di coordinate basato sulle coordinate locali di un oggetto. Le coordinate oggetto descrivono la posizione di un punto della geometria rispetto all'origine dell'oggetto stesso. Queste coordinate sono indipendenti dalla dimensione. Questo sistema di riferimento è però influenzato da fattore di scala dell'oggetto., quindi a un fattore non uniforme corrisponde una distorsione della texture che seguirà le trasformazioni dell'oggetto.



**Figura 5.5:** Coordinate object

### 5.4.5 Camera

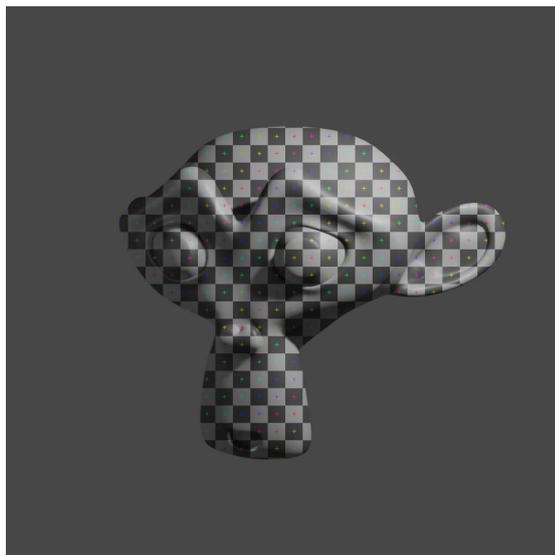
Le coordinate di tipo camera si riferiscono a un sistema di coordinate basato sulla vista della telecamera. Ciò significa che le coordinate sono basate sulla prospettiva della telecamera, associando un vettore che parte dall'oggetto fino all'origine della camera. Questo tipo di dati però non forniscono informazioni utili per oggetti che non sono statici, in quanto non si muoverebbero in modo solidale. Le coordinate di tipo Camera possono essere utilizzate per creare effetti speciali che dipendono dalla posizione della telecamera, come effetti di distorsione.



**Figura 5.6:** Coordinate camera

### 5.4.6 Window

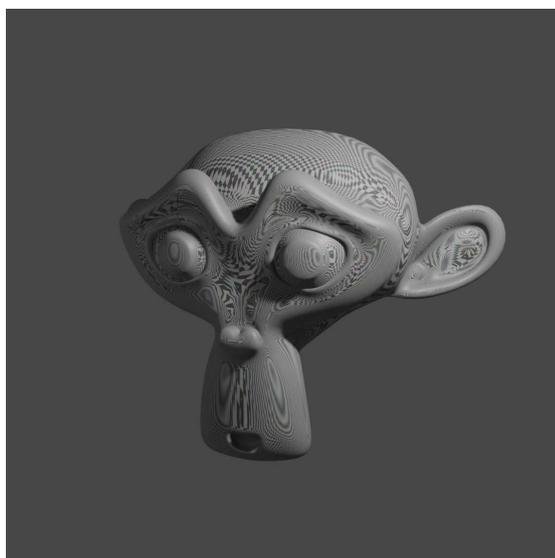
Questo sistema di coordinate è utilizzato per definire la posizione di un punto rispetto alla finestra di visualizzazione dell'utente. Il funzionamento è simile a quello delle coordinate screen space ma vengono ma in questo caso i valori vengono normalizzati in base alle dimensioni della scheda viewport. Il risultato è che la texture applicata sarà solidale a come la vista si muove nello spazio della scena.



**Figura 5.7:** Coordinate window

### 5.4.7 Reflection

Questo sistema di coordinate reflection sfrutta la riflessione della luce nell'oggetto per determinare la posizione dei punti. Si basa sul creare delle coordinate in base alla direzione del vettore di riflessione per i punti della mesh. Possono essere usati collegare una texture o materiale a delle zone di riflessione dell'oggetto.



**Figura 5.8:** Coordinate reflection

### 5.4.8 Tangent

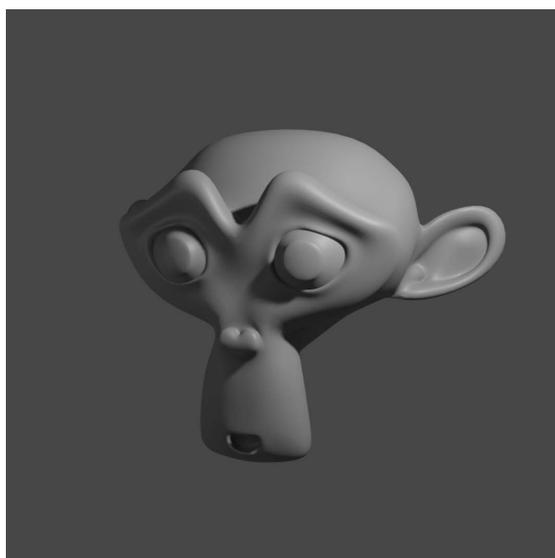
Queste coordinate non sono presenti nel nodo Texture coordinate ma in un nodo apposito con lo stesso nome. Può essere utilizzato, come le coordinate normal, per applicare effetti basati sulla direzione locale della superficie, come ad esempio effetti di rilievo o dettagli procedurali in uno shader. Infatti lo spazio tangent fornisce i vettori perpendicolari alla superficie nei punti della mesh.



**Figura 5.9:** Coordinate tangent

### 5.4.9 Position(Location)

Si riferiscono alle coordinate tridimensionali di un punto nello spazio rispetto all'origine del sistema di coordinate globale della scena. Queste coordinate rappresentano la posizione assoluta di un punto e possono essere utilizzate per vari scopi all'interno degli shader ma anche per l'utilizzo dei geometry node(in quel caso vengono chiamati location). Questo tipo di coordinate è quindi indipendente dalle trasformazioni ma solo dal punto in cui si trovano gli oggetti nella scena.



**Figura 5.10:** Coordinate position

#### 5.4.10 View vector

Questo tipo di coordinate si può ritrovare nel nodo Camera data e ha un funzionamento simile a quello camera: entrambi gli spazi usano il vettore che unisce un punto della geometria all'origine della camera ma nel caso del sistema di coordinate view vector questi vettori sono normalizzati. La natura di questo tipo di coordinate risulta in una texture piatta che vien proiettata sugli oggetti senza l'informazione di profondità.



**Figura 5.11:** Coordinate view vector

## 5.5 Setup dello shader

Lo schader che ho realizzato è il risultato di una combinazione di quello realizzato precedentemente per il progetto con alcune modifiche. Ho usato quindi lo shader precedente come shder base da cui estrapolare le informazioni colore e applicato in aggiunta gli effetti di outline, pattern a esagono, halftoning e retino a righe. Il sistema di coordinate usato è stato in ultima analisi quello UV per permettere di avere un risultato costante e non fastidioso durante le animazioni. Infatti usare degli spazi con un risultato più piatto come quello camera genera dei risultati non conformi al movimnto degli oggetti in scena.

## 5.6 Outline

Una feature grafiche dello stile che doveva essere progettato era quello di creare un outline attorno a i contorni delle figure. L'outline creato doveva ricreare il tratto grafico di un disegnatore andando a marcare uno dei tratti distintivi dell'animazione classica. Non potendo sfruttare il lavoro di illustratori che aggiungessero quest'effetto dopo il render della scena 3D si ci è dovuti concentrare su tecniche che fossero il più possibile automatiche. Per far ciò ho analizzato diversi metodi di per creare i contorni e si possono dividere principalmente in tre tipi: Freestyle, Line art modifier e Inverted hull. Il metodo precedentemente utilizzato era quello

del Freestyle ma dopo un'analisi si è deciso di optare per il metodo di inverted hull per motivazioni che illustrerò di seguito.

### 5.6.1 Metodo Freestyle

Freestyle è un motore di rendering non fotorealistico (NPR) basato su bordi/linee. Si basa su dati di mesh e informazioni sulla profondità Z per disegnare linee su tipi di bordi selezionati.



**Figura 5.12:** Soggetto ravvicinato con freestyle di spessore 1px



**Figura 5.13:** Soggetto lontano con freestyle di spessore 1px

Si può attivare Freestyle dal pannello proprietà di “Render” e attivando l’opzione apposita. Una delle opzioni messe a disposizione è quella di cambiare lo spessore della linea misurata in pixel, quindi la linea risulterà più sottile se la risoluzione dell’immagine viene impostata a valori maggiori e non quindi in modo proporzionale alle proporzioni dell’oggetto. Il resto delle opzioni sono editabili attraverso il pannello “Layer”([7]). Una delle categorie di opzioni che influenzano di più il risultato è “Edge type” che definisce in che modo definire le parti della geometria a cui aggiungere la linea di contorno. Un altro parametro importante è il “Crease angle” dell’“Edge detection” che determina il threshold dell’angolo per applicare l’outline sui contorni

Freestyle è un effetto effettuato in post-processing quindi non è possibile vederne gli effetti prima di aver effettuato un render. Come si può vedere dal grafico per prima cosa vengono prese le informazioni di rilevamento degli spigoli in base alle impostazioni selezionate e inserite in una view map. Dopo di che vengono

individuati quali tipi di linee e selezioni di line set verranno renderizzati in base alle impostazioni selezionate dall'utente e specificate come Line style options. Come ultimo passaggio Freestyle usa le opzioni di render per disegnare e renderizzare l'outline.

Dato che è calcolata dopo il render implica che se il modello ha una geometria molto complessa allora ci vorrà molto tempo per calcolare le linee di contorno.

I vantaggi offerti da questo metodo sono:

- La facilità di settaggio delle impostazioni
- Numero di opzioni di customizzazione del tratto
- Si applica all'intera scena

Gli svantaggi invece sono :

- Non è possibile visualizzare l'outline in preview dalla viewport
- Può richiedere molto tempo e memoria per essere calcolata, in quanto tutte le mesh vengono calcolate allo stesso momento

### 5.6.2 Line art modifier



**Figura 5.14:** Line art modifier

Per applicare il metodo di contorno effettuato con il Line art modifier bisogna raggruppare tutto ciò che si vuole modificare con l'outline dentro una stessa

collezione. Dopo aver fatto ciò si deve aggiungere, fuori dalla collezione, un oggetto “Grease pencil” di tipo “Blank” che fungerà da outliner. A questo “Grease pencil” si deve aggiungere un Line art modifier in cui si deve specificare la collezione sulla quale si desidera agire, oltre al layer e il materiale che si intende usare. Blender mette a disposizione le opzioni di Line Thickness e Opacity per modificare in maniera veloce lo spessore delle linee e l’opacità di queste. Da notare che lo spessore non è indicato in pixel ma è un valore proporzionale che va da 0 a 200. Un’altra caratteristica di questo metodo è che l’outline, sebbene sia possibile visualizzarlo in tempo reale sulla Viewport, è calcolato in base alla vista della camera selezionata, quindi porzioni del modello occlusi rispetto il punto di vista della camera non verranno tracciati.

Una tecnica per rendere questo metodo con uno stile grafica dal tratto meno rigido e meccanico si può aggiungere un modifier di tipo Subdivide di tipo simple e uno di tipo Noise. Regolando la Thickness, Strength e Noise scale del modifier di tipo Noise è possibile dare alle linee un tratto più incerto e meno meccanico.

I vantaggi offerti da questa tecnica sono:

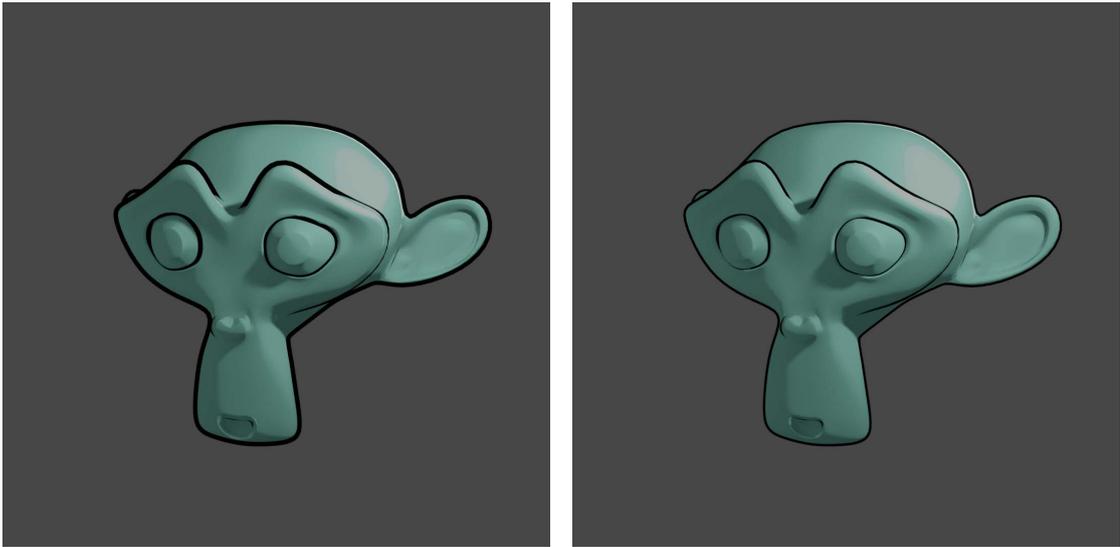
- Ha delle impostazioni facili da controllare
- È più veloce della tecnica Freestyle

Gli svantaggi invece sono :

- Il risultato visualizzato dipende sempre dalla posizione della camera
- Può causare dei ritardi di aggiornamento nella viewport

### 5.6.3 Inverted hull

Il metodo dell’inverted hull è realizzato utilizzando le proprietà dei materiali e il modifier di tipo solidify. Selezionando la mesh sulla quale si intende agire, si deve creare un nuovo materiale di tipo Emission con il colore voluto per il contorno, che nel nostro caso è il nero, e applicare nell’opzione Setting il Backface culling. La blend mode e la shadow mode da utilizzare sono di tipo opaco e none(per non creare conflitti e comportamenti anormali sulle self-shadow). Dopo aver creato il materiale che verrà usato si deve applicare il modifier Solidify alla mesh e agendo sulle normali in modo tale da selezionare le opzioni Flip e High quality. Tramite il parametro Offset, che riceve un numero, il modifier solidify realizza l’outline solo attraverso i materiali che hanno un numero maggiore o uguale come numero di stack. Dato che se viene selezionato un numero di offset maggiore al numero di materiali presenti il materiale scelto è l’ultimo, ho deciso di inserire un numero alto di 100 così da essere sicuro di selezionare il materiale dell’outline posizionato alla fine. Dato che il metodo di Inverted hull lavora direttamente sulla mesh, non per



**Figura 5.15:** Inverted hull spesso

**Figura 5.16:** Inverted hull poco spesso

esempio su un layer apposito, e in real time in modo indipendente alla camera, è possibile utilizzare proprietà come motion blur e la profondità di campo senza step aggiuntivi.

Questo metodo offre come vantaggi:

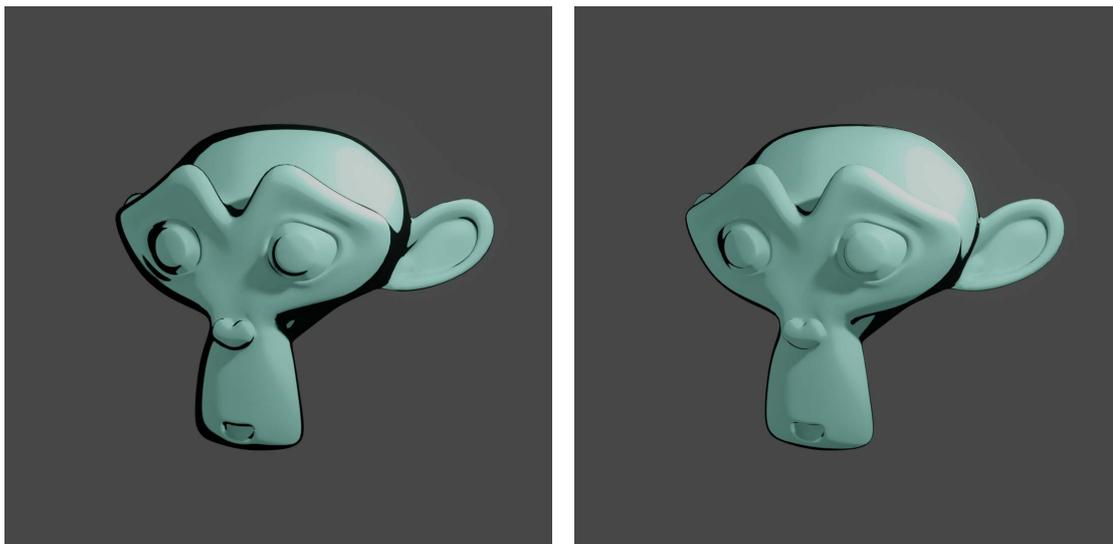
- Aggiornamento del risultato in tempo reale
- Applicazione diretta alla mesh, quindi c'è un'integrazione naturale con tutte le altre opzioni di Blender

Gli svantaggi che si possono riscontrare sono :

- Si deve applicare ad ogni oggetto individualmente
- Non lavora bene con tutte le geometrie
- Non è abbastanza responsivo alla direzione della luce

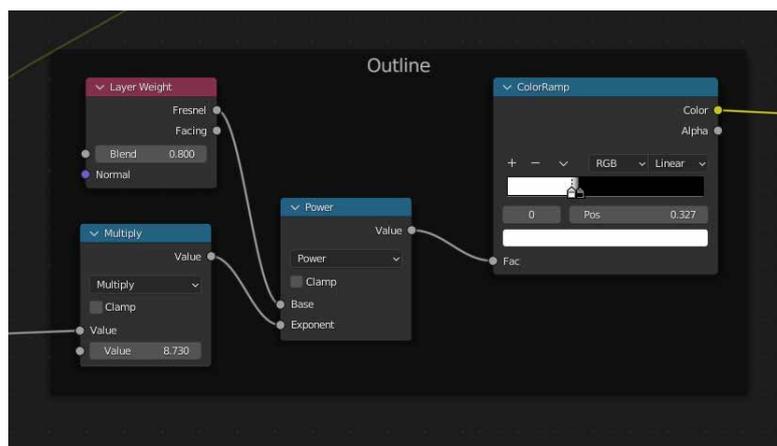
#### 5.6.4 Outline dinamico

Il metodo di inverted hull è quello che forniva i risultati migliori sia a livello artistico che di possibilità di modifica in real time sullo shader ma si è scelto di optare per un metodo integrato nello shader. L'obiettivo che volevo raggiungere con questa opzione era quella di creare un outline che reagisse e in modo dinamico in base alla luce della scena e che fosse facile da modificare in fase su shading sulle differenti mesh. Usiamo un Layer weight node di cui utilizzeremo l'output di Fresnel([8]) da



**Figura 5.17:** Outline dinamico spesso    **Figura 5.18:** Outline dinamico sottile

passare ad un nodo di color ramp. L'output di Fresnel serve a calcola quanta luce viene riflessa su un layer e attraverso il color ramp comprimeremo i valori per essere visibile l'effetto solo nei bordi della mesh. Per cambiare lo spessore dell'outline in base alla intensità di luce sull'oggetto dobbiamo dare un valore alla color ramp che dipenda da questa. Facendo passare il valore dell'intensità da lo shader di input creato in precedenza, facendolo passare attraverso il math node multiply per gestirne meglio il valore e unendolo all'output di Fresnel con un nodo math di tipo power possiamo dare al color ramp un valore dipendente dall'intensità della luce e che sia modificabile in base al parametro value del nodo multiply.



**Figura 5.19:** Nodi outline

I vantaggi offerti da questo metodo sono:

- Aggiornamento del risultato in tempo reale
- Applicazione diretta alla mesh, quindi c'è un'integrazione naturale con tutte le altre opzioni di Blender
- Responsivo alla direzione della luce, permettendo dei tratti più spessi nei punti in ombra

Gli svantaggi invece sono :

- Si deve applicare ad ogni oggetto individualmente

## 5.7 Pattern a esagono

Per la creazione dello shader, oltre ai classici effetti del pattern a righe e l'halftoning, si è provato ad esplorare una possibilità di fornire un tratto distintivo allacciandosi al motivo di un pattern esagonale. L'esagono è una figura che ritorna spesso nella serie e quindi si è cercato di esprimerla anche dal punto di vista del look del personaggio.

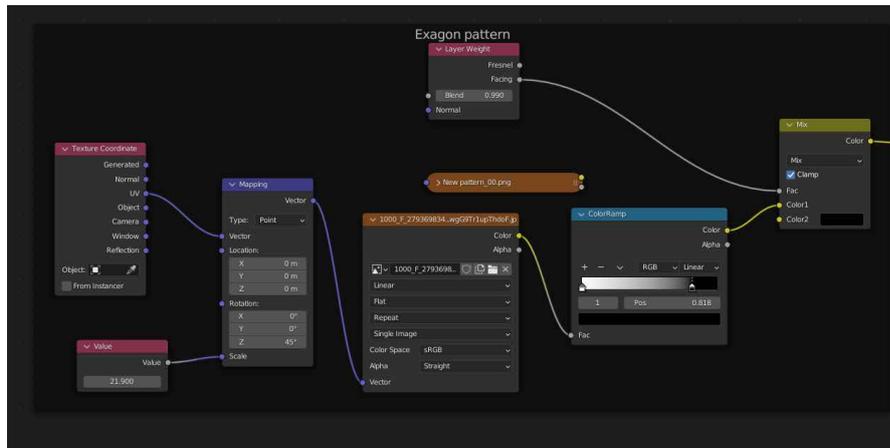


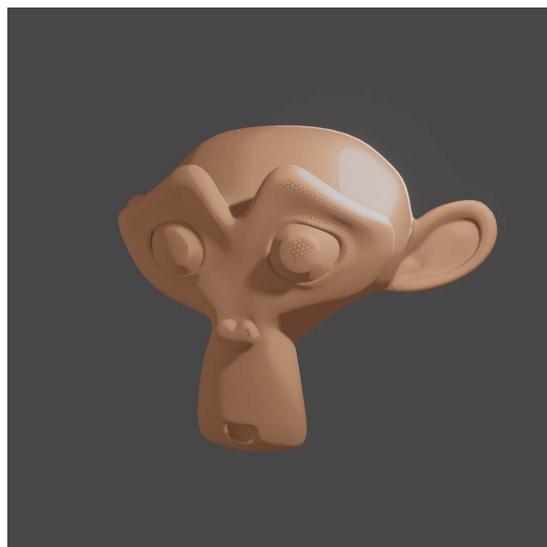
Figura 5.20: Nodi pattern a esagono

Per ricreare questo risultato mi sono servito di una texture seamless (un'immagine che può essere ripetuta senza che siano evidenti i confini tra le diverse ripetizioni) che rappresentasse un pattern a celle esagonale. Questo pattern è stato pilotato tramite un nodo di mapping per orientarlo a 45° ed un nodo Texture coordinate con coordinate UV. Si è scelto le coordinate UV perché producono meno distorsioni in fase di animazioni, l'unica accortezza è quella di fare un buon unwrap della

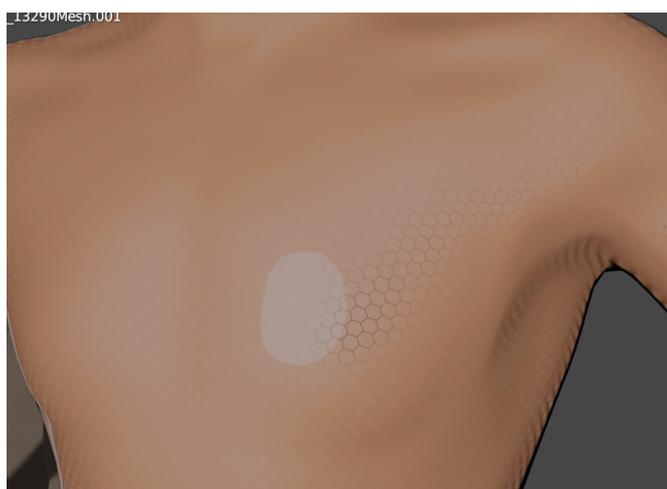
geometria in prima fase. La Image texture del pattern è stata modificato con un nodo Color ramp per deciderne il contrasto da applicare al modello e il risultato è stato unito con un nodo MixRGB che avesse come fattore di mix un input dato da un nodo Layer weight. Il nodo Layer weight da in output il valore di Facing che ci permette di inserire il pattern nei punti più riflessivi della mesh. L'output del MixRGB è poi stato usato come fattore di un Mix shader con in entrata lo shader base.



**Figura 5.21:** Pattern ad esagono coordinate camera



**Figura 5.22:** Pattern ad esagono coordinate UV

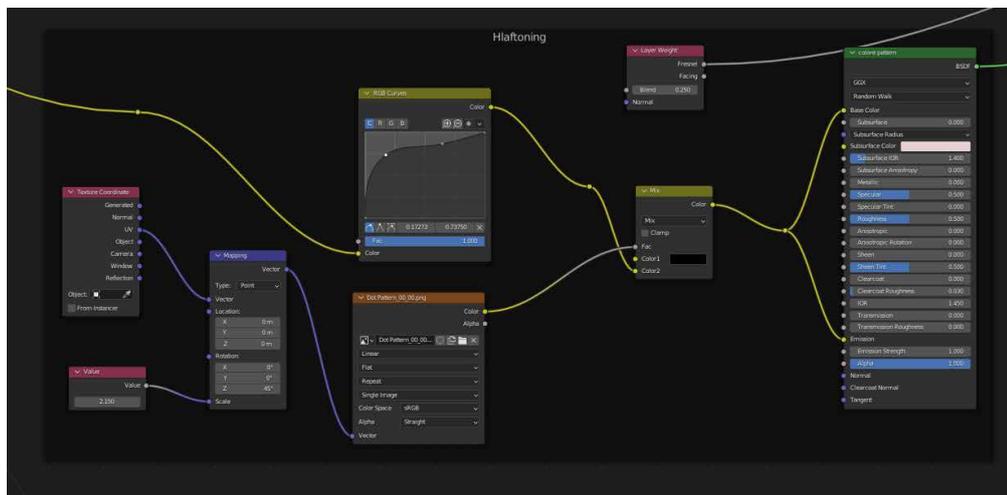


**Figura 5.23:** Pattern esagono

## 5.8 Halftoning

Per realizzare l'effetto halftoning si è sperimentato prima con un gruppo di nodi che utilizzavano un'immagine di texture e uno usando un nodo Voronoi (che genera un pattern a punti in modo procedurale).

Nel caso dell'uso di una Image texture, questa è stata pilotata in coordinate UV, come per il pattern ad esagono, e con un nodo di Mapping che desse un orientamento a  $45^\circ$  e ne regolasse la scala. L'output dell'Image texture è stato usato come fattore di blend in un nodo MixRGB che riceveva in ingresso il valore dello shader di base passato attraverso un RGB Curves per aggiustare in modo intuitivo l'intensità della luce alzando le ombre. Il risultato del MixRGB è stato messo in ingresso sia come Base color che Emission in un PrincipleBSDF shader per essere mixato tramite un Mix shader con lo shader di base e un fattore di blend dato da un nodo Layer weight. Il Layer weight è servito, tramite il parametro di Fresnel, a inserire l'effetto halftoning lungo i bordi dell'oggetto. Dopo questo



**Figura 5.24:** Nodi halftoning con immagine texture

tentativo si è provato ad usare un nodo Voronoi per generare il pattern a punti. Il gruppo di nodi segue una logica simile tranne nella generazione del pattern. L'output del nodo di Voronoi texture, pilotato in coordinate UV e inclinato di  $45^\circ$ , è stato messo in ingresso a un nodo Math Greater Than in modo tale da essere usato come threshold da comparare con il valore dello shader base. Il valore del nodo Math è stato messo in ingresso a un nodo MixRGB di tipo Multiply insieme al valore dello shader di base mixato in maniera analoga a quella precedente.

Seppure il secondo metodo sia procedurale l'effetto dato dalla texture offre un risultato più dettagliato e più facile da posizionare lungo solo i punti d'interesse del modello, ovvero i bordi, senza essere vistoso negli altre zone.

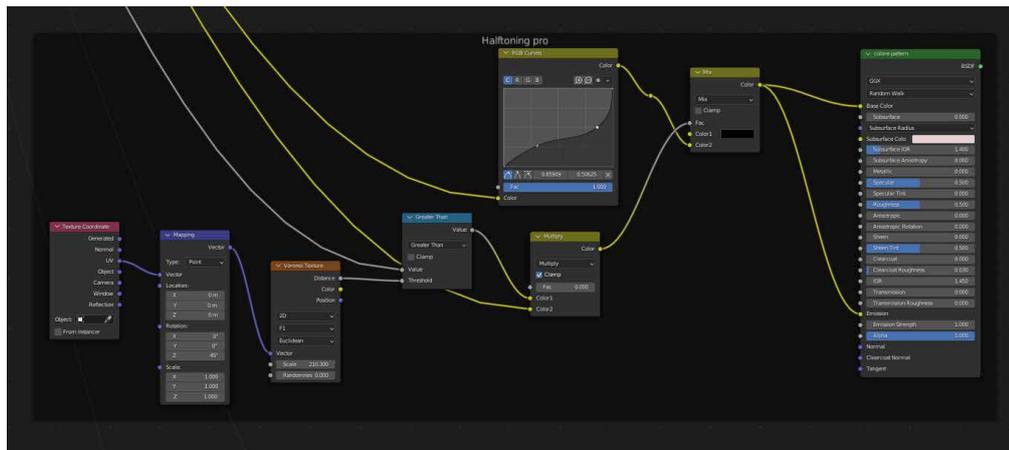


Figura 5.25: Nodi halfToning con Voronoi texture

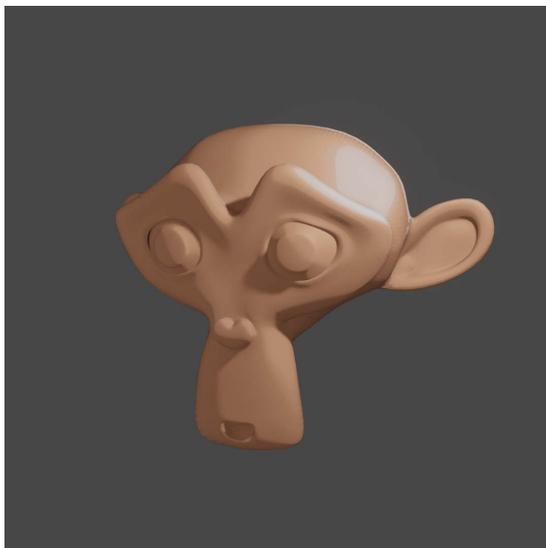


Figura 5.26: HalfToning in coordinate UV

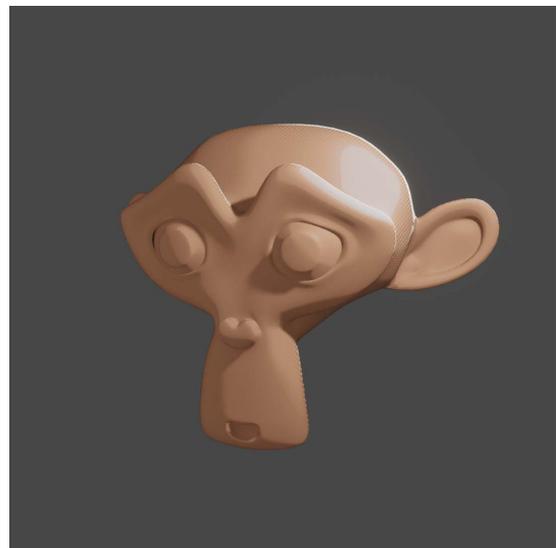


Figura 5.27: HalfToning in coordinate camera

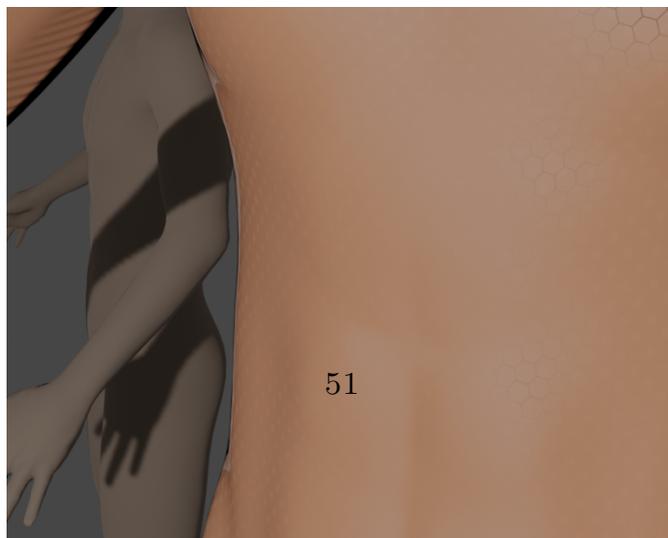
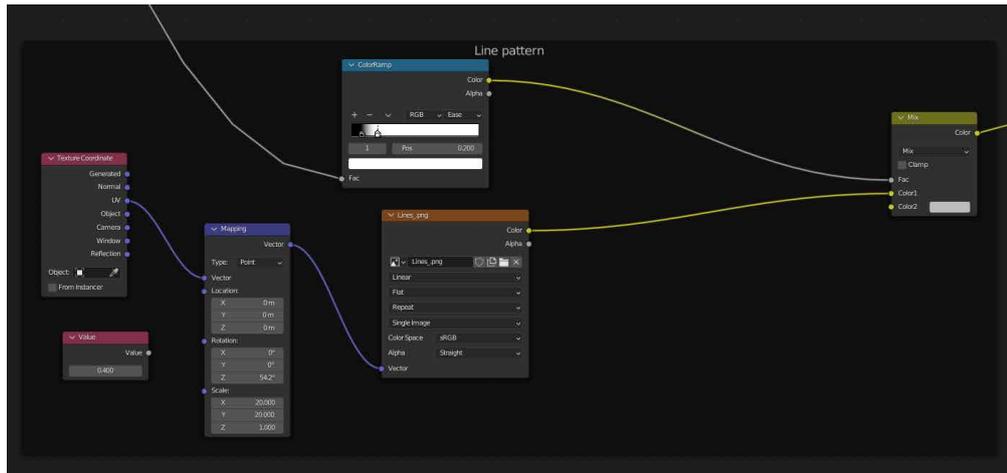


Figura 5.28: HalfToning

## 5.9 Retino a righe

Anche per l'effetto del retino a righe ho sperimentato sia con un immagine di texture che con un nodo texture 3D, ovvero Wave texture. Come ingresso alla



**Figura 5.29:** Nodi retino a righe con immagine texture

Image texture ho usato un nodo Mapping con una inclinazione sull'asse Z di circa  $50^\circ$  che usa coordinate UV. L'output di questo nodo è stato inserito in un nodo di MixRGB che ha come fattore di mix il valore dello shader di base filtrato attraverso un nodo ColorRamp. Il nodo Color Ramp permette di apporre il pattern a righe solamente nei punti in ombra del modello. L'output del MixRGB è stato usato come fattore di mix in un nodo Shader che ha in entrata l'output shader del gruppo di nodi dell'halftoning.

La prova con il nodo Wave texture è simile come logica al tentativo precedente. Al posto di usare un'immagine di texture ho usato il nodo Wave texture in coordinate UV e con un'inclinazione di  $45^\circ$ , data al nodo Texture coordinate e Mapping. L'output color del Wave texture è stato usato come ingresso colore nel mixRGB e il risultato di quest'ultimo è stato usato come fattore di mix nel Mix shader successivo. In questo caso i risultati sono molto simili ma il metodo con l'immagine Texture offre un risultato più costante e senza deformazioni lungo il modello dell'oggetto in questione.

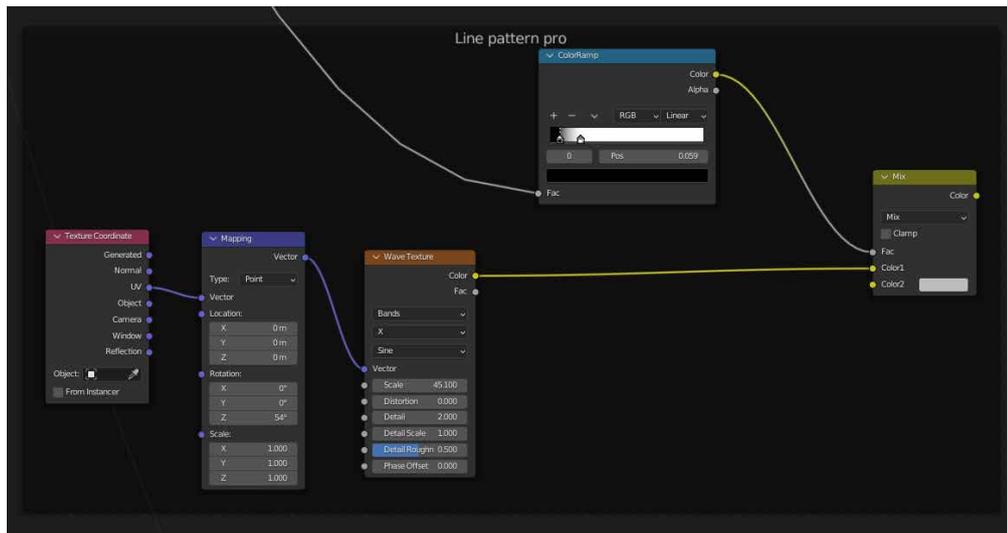


Figura 5.30: Nodi retino a righe con Wave texture



Figura 5.31: Retino a righe in coordinate UV

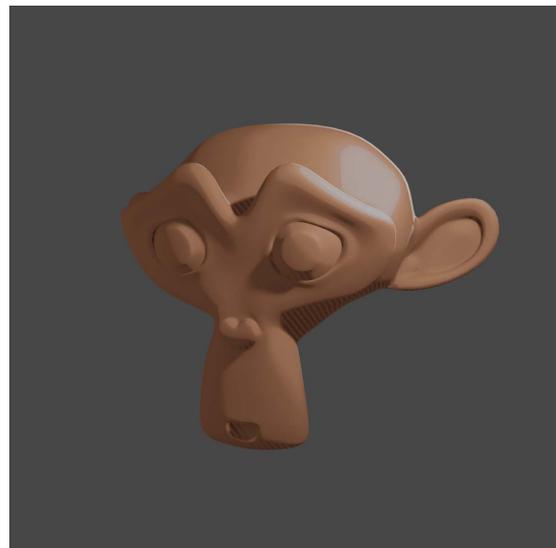


Figura 5.32: Retino a righe in coordinate camera

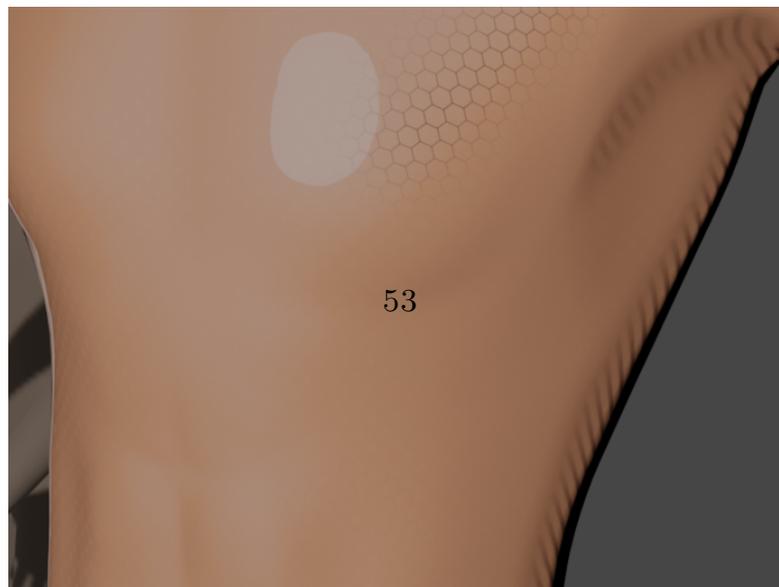


Figura 5.33: Retino a righe

## 5.10 Ologramma



**Figura 5.34:** Ologramma

Per la realizzazione della scena del night club, in fase di preproduzione, abbiamo scelto, in base ai concept realizzati, di inserire un ologramma di un volto umano dentro lo spazio del locale. Per realizzare questo modello ho sviluppato uno shader apposito in grado di cambiare i parametri nel tempo e quindi fornire un'animazione nella scena. Ho usato un noise texture in coordinate generated e un nodo ColorRamp per generare l'input di IOR(index of refraction) di un nod Fresnel per creare un effetto a righe orizzontali. Questo output è stato mixato con quello di un altro nodo Fresnel, attraverso un nodo MixRGB, per accentuare l'effetto nei punti di maggior riflesso. Il risultato di questi nodi è servito come fattore di mixing per un mix shader che riceveva come input un emission shader con il colore dell'ologramma. Per effettuare delle animazioni di disturbo nell'ologramma ho inserito delle curve di noise per il valore di strength nel nodo emission e lo IOR del nodo Fresnel. Per effettuare invece il movimento delle linee in verticale ho applicato una curva lineare al valore Y del nodo di Mapping.

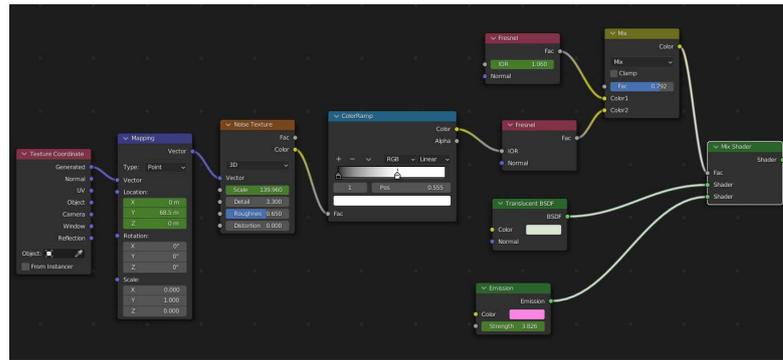


Figura 5.35: Nodi shader ologramma

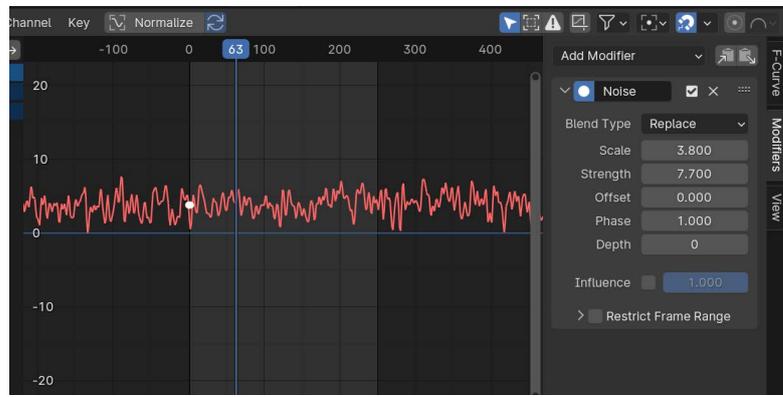


Figura 5.36: Grafico dell'effetto noise sul valore strength dell'emission nel tempo

## 5.11 Prove

### 5.11.1 Normal paint strokes

Una possibile tecnica che abbiamo cercato di esplorare è stata quella definita come paint strokes. L'idea è quella di dare una estetica da pennellate evidenti per dare un tocco distintivo alle immagini della serie. Il concetto alla base di questo metodo consiste nell'estrapolare le informazioni sulle normali di una mesh attraverso un bake delle normali. Effettuato il bake, si può utilizzare l'immagine della mappa delle normali, che contiene le informazioni dei vettori tangenti in base al colore dell'immagine, e modificarne l'apparenza per creare l'effetto desiderato. Con un qualsiasi software di illustrazione si può prendere un colore in una zona della mappa e usarlo in modo uniforme per colorare con un brush in una zona nell'intorno del punto da cui abbiamo estratto il colore. In questo modo creiamo un appiattimento delle normali in quella zona a valori molto simili e restituiamo un senso di pennellata in quelle aree. Finita la modifica della mappa delle normali,

questa può essere usata in Blender tramite un nodo Image texture e uno normal map, per trasferire le informazioni sulle normali al nodo di shader. Il risultato di questo nodo, in cui inseriamo pure l'informazione colore dello shader base è mixato con un mix shader del colore base.

Provando più tipi di mixing con lo shader principale (che usa outline, retino a righe, pattern ad esagono e retino a righe) il risultato è un po' confusionario, senza tralasciare il fatto che il passaggio da effettuare su software di illustrazione è un passaggio ulteriore e buon risultato dipende dall'artista che effettua il lavoro.



**Figura 5.37:** Effetto paint brush

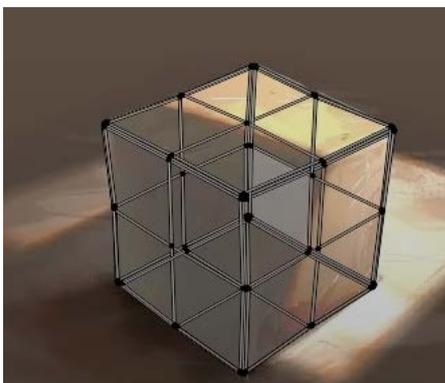
### 5.11.2 Camera projection

L'idea alla base del processo con camera projection è di usare lo shader in blender come base della texture finale, che dovrà essere finalizzata con ulteriori dettagli in un software di illustrazione e rimappata su una texture creata attraverso una camera projection. Il motivazione dietro questo metodo è di aggiungere dettaglio ma anche ombreggiature drammatiche che restituiscano un valore artigianale al risultato. Anche se questi dettagli non si muovono in modo realistico alla luce nella scena è un risultato piacevole che è stato usato per esempio in una serie tv come Arcane.

Per prima cosa bisogna selezionare l'inquadratura da cui si vuole riprendere il modello, perché le successive modifiche avranno senso solo da una certa angolazione e la proiezione finale dovrà essere effettuata con lo stesso punto di vista. Effettuato il render, dalla posizione che scegliamo, lo usiamo per aggiungere dettagli nel

software di illustrazione ed effettuiamo un unwrap della texture del tipo project from view. Passando all'editor di shading inseriamo la texture modificata come Image texture prendendo come input l'unwrap che abbiamo effettuato. Se vogliamo effettuare altre proiezioni allo stesso oggetto bisogna effettuare le stesse operazioni ma usare due unwrap differenti nello stesso material per mantenere entrambe le operazioni.

Questa tecnica, seppur possa aumentare l'effetto drammatico della scena, comporta un ulteriore lavoro che richiede più fasi e ha bisogno di una figura apposita per essere effettuato. Per questi motivi non ha fatto parte del nostro workflow se non come tool aggiuntivo per aggiungere dettagli se necessari.



**Figura 5.38:** Unwrap UV della proiezione

# Capitolo 6

## Lighting

L'illuminazione di una scena digitale è un elemento fondamentale che conferisce realismo e atmosfera alle immagini digitali. Essa gioca un ruolo cruciale nella creazione di scene visive tridimensionali, influenzando l'aspetto e la percezione dei modelli virtuali. Per illuminare la scena abbiamo scelto il software Blender e, come già detto precedentemente, il motore di rendering Eevee. La scelta di usare questo software è legata anche allo sviluppo dello shader nella stessa piattaforma e ha significato impostare la scena e le luci secondo impostazioni proprie del motore di rendering usato. Nell'illuminare una scena in computer grafica si applicano le stesse tecniche e teorie che verrebbero usate in una produzione live action come la scelta della giusta sorgente luce, la sua intensità e temperatura. Durante questo capitolo introdurremo le teorie strumenti utilizzati per illuminare la scena e le scelte artistiche riguardanti colore e atmosfera.

### 6.1 Teorie e tecniche dell'illuminazione

L'illuminazione di una scena deve tenere conto della natura della sorgente luminosa e come interagisce con le superfici. Gli ambienti di grafica virtuale sono stati sviluppati in modo tale da offrire degli strumenti che si avvicinino quelli reali, in modo tale da usare le stesse logiche e parametri ai quali i direttori delle fotografia sono abituati a gestire nel mondo reale. Simulare una scena in grafica digitale significa gestire delle fonti luminose e se queste si comportano come si comporterebbero in uno scenario reale è più facile costruire delle inquadrature illuminate bene.

I parametri principali da gestire per l'illuminazione sono:

- **La temperatura:** La temperatura del colore della luce è un concetto chiave nell'illuminazione e si riferisce al colore apparente della luce emessa da una sorgente luminosa. Questo concetto è misurato in gradi Kelvin (K), rispecchiando la natura fisica delle sorgenti luminose in relazione alla loro temperatura e

calore. La temperatura del colore influenza notevolmente l'aspetto visivo di una scena, determinando se la luce è percepita come calda, neutra o fredda. Sorgenti luci a temperature basse, che vanno da circa 2700 K a 3500 K, offrono una luce calda che è associata ai toni gialli, arancioni e rossi. Crea un'atmosfera accogliente e intima ed è spesso utilizzata per simulare la luce del tramonto o delle lampade a incandescenza. calda . La luce neutra è invece emessa da sorgenti con una temperatura fra circa i 3500 K e i 5.000 K. E comunica una luce equilibrata e può comunicare approssimativamente la luce del giorno, o del tramonto. Questa temperatura del colore è spesso utilizzata in contesti in cui si desidera una resa dei colori accurata e una percezione naturale della luce. Fra circa i 5.000 K e i 10.000 K si può parlare di luci fredde che sono associate al tono del bianco e dell'azzurro e vengono spesso usate per simulare la luce del giorno(5600 K) in condizioni di luce intensa o illuminazione artificiale fluorescente. Nell'illuminazione cinematografica e fotografica, la scelta della temperatura del colore è una decisione creativa importante che influisce sull'umore e sull'estetica della scena. Può essere utilizzata per trasmettere emozioni, suggerire l'ora del giorno e contribuire alla coerenza visiva di un film o di una serie. I professionisti spesso sperimentano con la temperatura del colore per ottenere effetti specifici e raggiungere l'atmosfera desiderata nella loro opera.

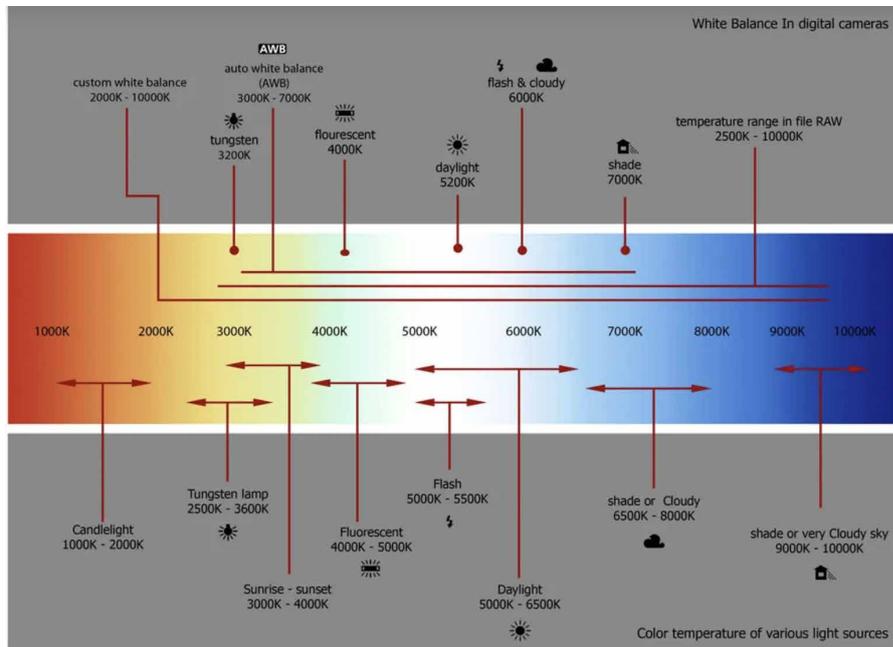


Figura 6.1: Temperatura colore

- **Colore:** creare una luce colorata su un set significa creare una luce con uno spettro completo (bianca), alla quale vengono sottratte determinate frequenze. Spesso questo viene creato tramite l'uso di filtri applicati alle sorgenti luminosi. Riprendere una luce colorata può inoltre essere effettuato tramite l'apposizione di filtri colori nell'obiettivo della camera (filtrando le componenti non desiderate) oppure in postproduzione cambiando la natura di una particolare luce in un'altra.
- **Intensità della luce :** L'intensità della luce in ambito cinematografico è una grandezza che viene messa in relazione alla potenza di una sorgente luminosa e la sua distanza dall'oggetto di riferimento. La potenza di una sorgente si misura in Watt ed è un elemento fondamentale nel determinare l'impatto della luce nella scena da mettere in relazione alla distanza dal soggetto ripreso nell'inquadratura. Si può usare per creare un'illuminazione brillante ed evidente o creare atmosfere più soffuse o per evidenziare specifici dettagli senza generare una forte luminosità generale. L'intensità di una luce in un punto è inversamente proporzionale al quadrato della distanza dalla sorgente.
- **Luce dura o morbida:** Una luce dura crea delle ombre con contrasti evidenti rispetto a una morbida che genera ombre meno definite con contorni sfumati. La distanza ravvicinata al soggetto produce un'illuminazione delicata e diffusa, mentre una distanza maggiore genera un'illuminazione intensa con ombre ben delineate.
- **Concentrata o diffusa:** La dimensione della sorgente impatta il tipo di luce che si viene a creare. Una luce concentrata è data da dimensioni ridotte della sorgente e provoca un effetto di luce più duro, mentre dimensioni più ampie della sorgente creano un effetto diffuso che risulta in una luce progressivamente più morbida.

Definire come posizionare le luci in relazione al soggetto e l'una all'altra è una componente importante del lavoro di illuminazione di un'inquadratura. Il setup più utilizzato è quello del sistema a tre punti luce, che è composto da una key light, una fill light e una back light, al fine di illuminare il soggetto in maniera completa. La key light è la principale sorgente di luce e solitamente è posizionata frontalmente o lateralmente al soggetto. Questa luce definisce la forma del soggetto e determina la direzione generale dell'illuminazione. La fill light è posizionata sul lato opposto della chiave per ridurre e ammorbidire le ombre create dalla luce principale. Di solito il lato più in ombra di un volto è quello rivolto verso la camera. La back light invece è posizionata dietro il soggetto per illuminare il contorno del soggetto creare distacco fra il questo e lo sfondo. A partire da questo setup molto canonico i direttori della fotografia cercano di elaborare le relazioni fra queste sorgenti per creare

il risultato drammatico voluto. Un esempio di variazione da questo setup è per esempio la Rembrandt light (dal nome del faso pittore fiammingo), che sfrutta luce frontale a tre quarti che comunque illumini lo zigomo della parte di viso in ombra, creando un gioco di chiaroscuri e formando triangolo di luce sotto l'occhio opposto alla luce principale.

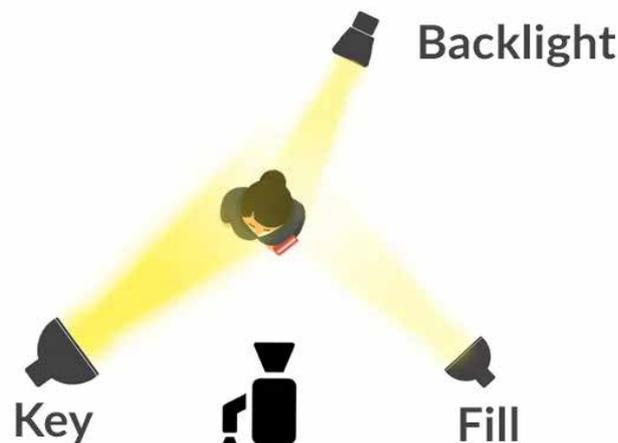


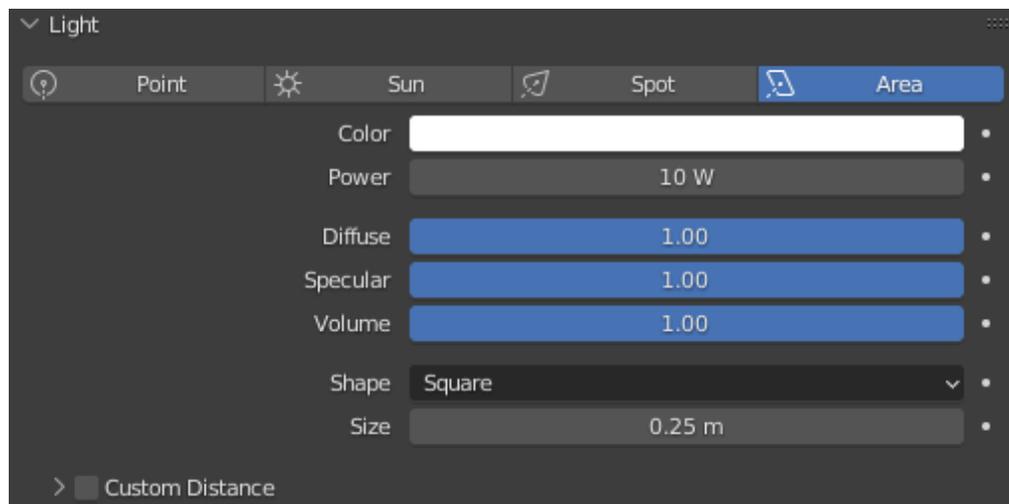
Figura 6.2: Three point lighting

## 6.2 Ambiente Blender

In questo capitolo affrontiamo come l'ambiente software di Blender permetta di illuminare una scena 3D([9]). La piattaforma offre la possibilità di usare 4 tipi di luci:

- Point : è una sorgente di luce che emette luce in tutte le direzioni da una sorgente puntiforme. Può essere utilizzata come modello semplice, ad esempio, di una lampadina. Ne si può regolare la potenza per incrementare l'intensità della luce e il raggio per modificare la natura puntiforme della sorgente in una sferica. Un raggio maggiore implica ombre più morbide e punti luce speculari, apparendo meno intense a parità di potenza.
- Spot: emette luce in una direzione specifica, creando un fascio di luce simile ad un cono. Come la point light si può modificare la potenza e il raggio con effetti uguali. Inoltre si può modificare la grandezza(size) del cono indicata in un angolo in gradi (da 1 a 180) e il parametro blend che controlla l'angolo del cono interno che determina i punti dai quali la luce diminuisce d'intensità fino al cono esterno.

- Area: è una sorgente che emette luce in tutte le direzioni da una superficie piatta, creando una luce diffusa. Si può modificare la potenza e il tipo di superficie emittente fra (rettangolo, quadrato, disco e elisse), con la possibilità di controllare le relative dimensioni.
- Sun: emette luce globale dalla direzione specificata del vettore direzione indicata tramite la modifica di un angolo nel pannello proprietà o modificando l'angolazione del raggio dalla schermata viewport. Il punto dove è posizionata la luce non influenza il risultato in quanto i raggi sono considerati tutti paralleli.
- Emissiva: questa sorgente non è elencata fra le luci che si possono inserire ma è creata tramite un oggetto a cui viene applicato un materiale con un nodo emission. Con l'aggiunta di questo nodo tutta la mesh dell'oggetto diventa una luce diffusa emissiva di cui puoi controllare colore, potenza.



**Figura 6.3:** Luci di blender

Per tutte le luci è possibile modificarne il colore cliccando il settaggio apposito. Possiamo inserire valori specifici per RGB(componente rossa, verde, blu), HSV(componente di tinta, saturazione e valore intensità) o Hex(il codice esadecimale del colore) se desideriamo ottenere un colore preciso. C'è anche uno strumento contagocce per selezionare il colore da un punto qualsiasi nella nostra vista. Nelle luci reali è più solito controllare solo la temperatura colore di una luce che consiste, nell'ambiente Blender, nel selezionare i colori della color wheel lungo la diagonale in figura.

Le sorgenti luce di blender in Eevee invece di indicare il massimo di rimbalzi della luce offrono tre parametri reflection multiplier per gestire l'interazione della

luce con l'ambiente. Valori più alti di questi parametri renderanno le componenti di riflessione diffusa, speculare e volume più riflessive per la luce in questione.



**Figura 6.4:** Temperatura colore in Blender

È stato inoltre necessario modificare alcuni parametri([10]) per lavorare meglio con il motore di render di Eevee. Uno di questi è stato la risoluzione delle ombre nel pannello delle proprietà di rendering, sia per fonti come la luce sun che le altre, che sono state impostate a 4096 px. Abilitando l'opzione soft shadow Eevee tratta le luci con un raggio maggiore come diverse luci puntiformi in posizioni differenti mischiando poi i risultati di queste ombre in una più morbida. Un'impostazione che era necessario attivare, anche per il tipo di ambientazione, è stato quella del bloom che rende visibile l'effetto glow della luce nell'ambiente. Per ogni luce ho modificato il parametro di bias delle ombre ad un valore di circa 0.001 per far sì che non si creassero artefatti nella proiezione delle ombre di oggetti su degli altri.

## 6.3 Reference

A partire dalla documentazione che avevamo a disposizione come team di lavoro, come la sceneggiatura e gli sketch dei concept, abbiamo cercato delle reference che potessero dare un'idea dell'ambiente descritto. La direzione sulla quale abbiamo ricercato immagini di riferimento era quella di un locale notturno con tante luci neon e led dai toni futuristici e cyber punk. Le refernce, come già scritto erano essenziali per dare una direttiva unica durante tutte la fasi del progetto ma è stata anche cruciale per capire come illuminare l'ambiente. L'ambiente è completamente

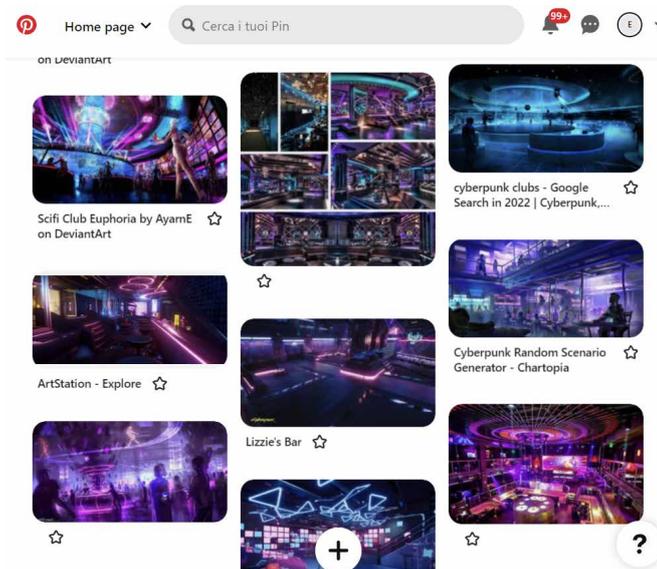


Figura 6.5: Reference



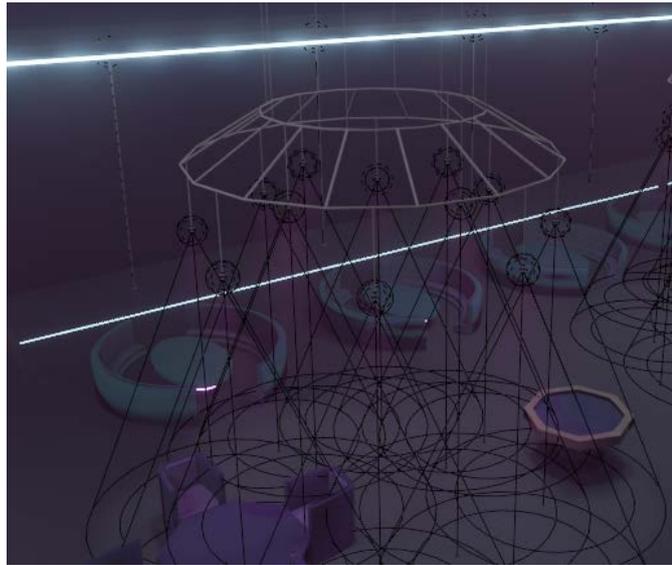
Figura 6.6: Palette colori

interno quindi abbiamo deciso di distaccarci dalla palette di colori caratteristica della serie, che è più tendente ai toni crepuscolari che vanno dall'arancione al rosso, ma di concentrarci su una palette più tipica del genere cyber punk. Nello scegliere i toni abbiamo deciso di puntare su dei toni analoghi che quindi andassero dall'azzurro al violetto.

## 6.4 Setup luci nella scena

Per la scena in questione volevo creare un ambiente tipo da nightclub, quindi con toni molto scuri scuri ma contrastati da luci colrate, secondo la palette che avevamo scelto. Per dare un tono caratteristico dell'ambientazione cyberpunk abbaio inserito

tanti neon per dare dettaglio. In questo caso il materiale emission applicato ci dava la possibilità di rendere visibile il bool, l'effetto di tipo glow attorno alla sorgente di luce, e dare più atmosfera all'ambiente. Per giustificare diegeticamente alcune luci emissive, che non producevano abbastanza luce, ho inserito delle area light in alcuni punti per dare un una luce diffusa di base per illuminare l'ambiente. Nel creare le strutture centrali del lampadario sono stati inserite delle spot light, ma anche queste sono state rinforzate da area light per dare un contributo ambientale maggiore.



**Figura 6.7:** Luci ambiente(1)



**Figura 6.8:** Luci ambiente(2)



**Figura 6.9:** Luci ambiente(3)



**Figura 6.10:** Luci ambiente(4)

# Capitolo 7

## Riultati e conclusioni

### 7.1 Risultati

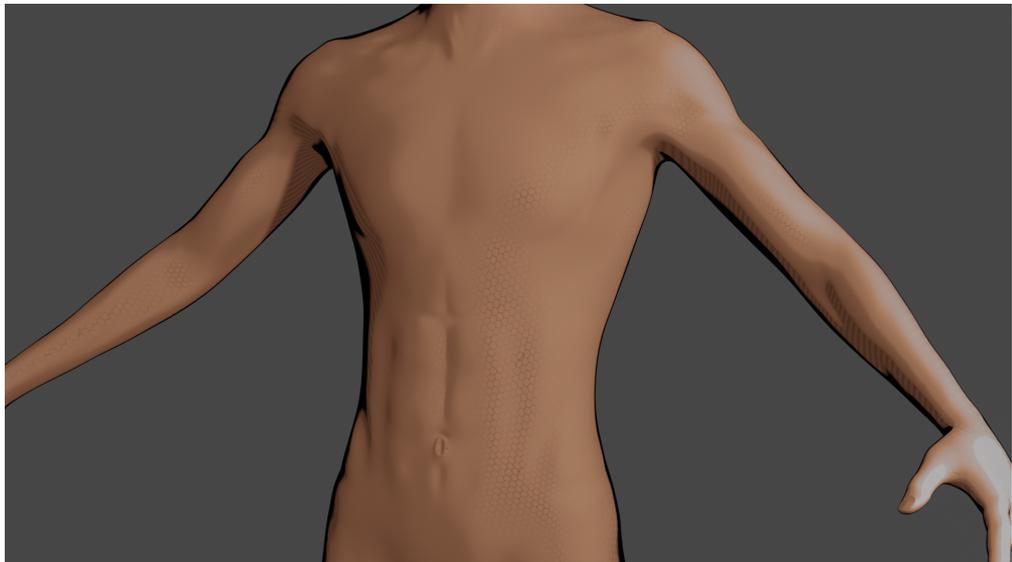
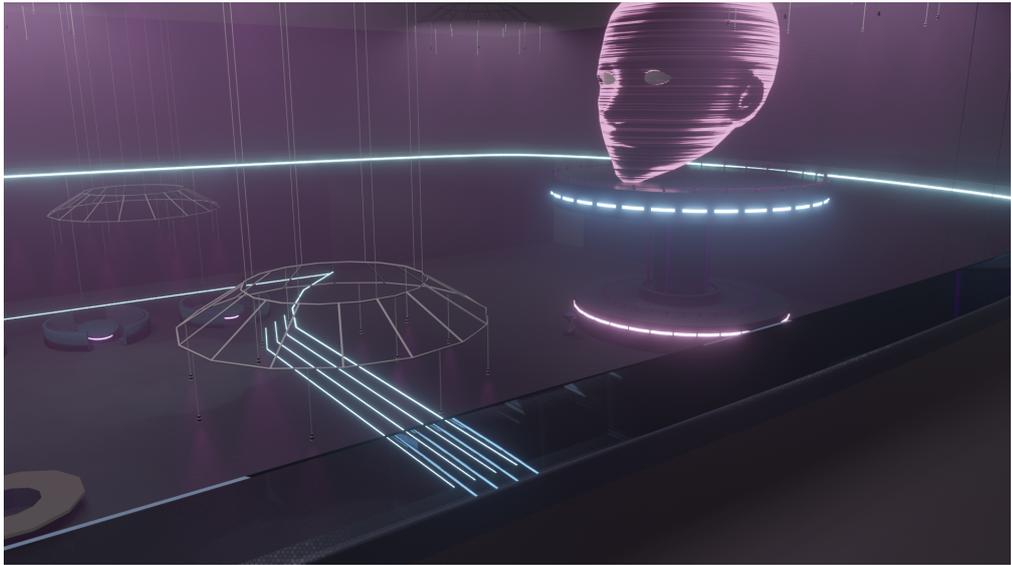


Figura 7.1



**Figura 7.2**



**Figura 7.3**



Figura 7.4

## 7.2 Conclusioni

Analizzando i risultati presenti in questo capitolo possiamo tirare le somme sulle performance dello shading realizzato dal punto di vista del risultato estetico, di sviluppo, facilità di utilizzo all'interno del workflow di lavoro e adattabilità all'animazione.

La possibilità di adattare le varie caratteristiche di questo shader alle varie esigenze di ogni modello e ogni contesto da la flessibilità a chi deve applicare le texture di realizzare il risultato voluto. Ciò implica una certa adattabilità ai vari contesti, scene e oggetti che sono da realizzare. Il risultato finale si avvicina ai casi studio nell'ottica di offrire un'estetica da fumetto.

Per come è stato strutturato, lo shader offre la possibilità di modificare il risultato in modo agevole e modulare. Dividendo la pipeline di shading in diverse sezioni indipendenti è possibile agire in diverse situazioni in modo indipendente. Per quanto riguarda le coordinate di applicazione della texture quella, uv è quella che offre i risultati più conformi per evitare effetti fastidiosi in fase di animazione. Seppur questo spazio di coordinate si debba adattare in modo preciso al modello offre una facilità di uso che lo spazio camera non offre. D'altro canto, in contesti in cui ci sono poche ombre e l'effetto dello shader è meno preponderante, le coordinate di tipo camera possono essere utilizzate e fornire un look più piatto, tipico del mondo del fumetto.

Per quanto riguarda dei risvolti futuri, potrebbe essere interessante effettuare un effetto paint stroke agendo in procedurale per velocizzare un processo che potrebbe

essere lungo da effettuare. Un'area di sviluppo può essere quella dell'integrazione nel processo di un digital artist per aggiungere dettaglio allo shading con la tecnica della camera projection.

In base ai test effettuati si può ritenere che gli strumenti forniti attraverso questo lavoro su vari aspetti dell'estetica del progetto abbiano raggiunto il livello prefissato all'inizio della lavorazione. Nel contesto di un progetto che dovrà comunque continuare, i tool sviluppati permettono una continuazione indipendente del progetto con la possibilità di adattarlo ai lavori futuri ed essere successivamente migliorato.

# Bibliografia

- [1] «ReveriePressbook». In: (). URL: <https://robin.studio/reverie/ReveriePressbook.pdf> (cit. a p. 1).
- [2] «Image Work - Spider-Man™: Into the Spider-Verse». In: (). URL: <https://www.imageworks.com/our-craft/feature-animation/movies/spider-man-spider-verse#:~:text=In%20order%20to%20achieve%20the,every%20one%20second%20of%20film.> (cit. a p. 5).
- [3] «Blender». In: (). URL: <https://www.blender.org/> (cit. a p. 10).
- [4] «Storia di Blender». In: (). URL: [https://docs.blender.org/manual/en/3.4/getting\\_started/about/history.html](https://docs.blender.org/manual/en/3.4/getting_started/about/history.html) (cit. a p. 10).
- [5] «Rendering in Blender». In: (). URL: <https://docs.blender.org/manual/en/3.4/render/introduction.html> (cit. a p. 11).
- [6] «Motion Blur di Eevee». In: (). URL: [https://docs.blender.org/manual/en/3.4/render/eevee/render\\_settings/motion\\_blur.html](https://docs.blender.org/manual/en/3.4/render/eevee/render_settings/motion_blur.html) (cit. a p. 19).
- [7] «Opzioni Freestyle Line set - Blender». In: (). URL: [https://docs.blender.org/manual/en/latest/render/freestyle/view\\_layer/line\\_set.html](https://docs.blender.org/manual/en/latest/render/freestyle/view_layer/line_set.html) (cit. a p. 43).
- [8] «Fresnel node - Blender». In: (). URL: [https://docs.blender.org/manual/en/latest/render/shader\\_nodes/input/fresnel.html](https://docs.blender.org/manual/en/latest/render/shader_nodes/input/fresnel.html) (cit. a p. 46).
- [9] «Light objects - Blender». In: (). URL: [https://docs.blender.org/manual/en/latest/render/lights/light\\_object.html#common-settings](https://docs.blender.org/manual/en/latest/render/lights/light_object.html#common-settings) (cit. a p. 61).
- [10] «Light settings - Blender». In: (). URL: <https://docs.blender.org/manual/en/latest/render/eevee/lighting.html> (cit. a p. 63).

# Elenco delle figure

1.1	Reverie cover . . . . .	2
1.2	Concept art . . . . .	2
1.3	Stanza Nadya . . . . .	4
2.1	Spider-Man: Into the Spider-Verse . . . . .	6
2.2	Love, death and robots - L'erba alta . . . . .	7
2.3	Arcane . . . . .	8
3.1	Illuminazione globale - Cycles e Eevee (CGCOOKIE.COM) . . . . .	13
3.2	Riflessione - Cycles e Eevee (CGCOOKIE.COM) . . . . .	14
3.3	Rifrazioni - Cycles e Eevee (CGCOOKIE.COM) . . . . .	14
3.4	Ambient occlusion - Cycles e Eevee (CGCOOKIE.COM) . . . . .	15
3.5	Ombre - Cycles e Eevee (CGCOOKIE.COM) . . . . .	16
3.6	Luce volumetrica - Cycles e Eevee (CGCOOKIE.COM) . . . . .	17
3.7	Subsurface scattering - Cycles e Eevee (CGCOOKIE.COM) . . . . .	18
3.8	Profondità di campo - Cycles e Eevee (CGCOOKIE.COM) . . . . .	18
3.9	Motion blur([6]) . . . . .	19
3.10	Bloom . . . . .	20
4.1	Character sheet . . . . .	22
4.2	Sketch night club . . . . .	22
4.3	Documenti riproduzione . . . . .	22
4.4	Breakdown spoglio sceneggiatura . . . . .	23
4.5	Storyboard . . . . .	25
4.6	Animatic . . . . .	27
4.7	Gantt . . . . .	29
5.1	Shader editor . . . . .	31
5.2	Coordinate generated . . . . .	34
5.3	Coordinate normal . . . . .	35
5.4	Coordinate UV . . . . .	36
5.5	Coordinate object . . . . .	37

5.6	Coordinate camera . . . . .	38
5.7	Coordinate window . . . . .	39
5.8	Coordinate reflection . . . . .	39
5.9	Coordinate tangent . . . . .	40
5.10	Coordinate position . . . . .	41
5.11	Coordinate view vector . . . . .	42
5.12	Soggetto ravvicinato con frestyle di spessore 1px . . . . .	43
5.13	Soggetto lontano con frestyle di spessore 1px . . . . .	43
5.14	Line art modifier . . . . .	44
5.15	Inverted hull spesso . . . . .	46
5.16	Inverted hull poco spesso . . . . .	46
5.17	Outline dinamico spesso . . . . .	47
5.18	Outline dinamico sottile . . . . .	47
5.19	Nodi outline . . . . .	47
5.20	Nodi pattern a esagono . . . . .	48
5.21	Pattern ad esagno coordinate camera . . . . .	49
5.22	Pattern ad esagno coordinate UV . . . . .	49
5.23	Pattern esagono . . . . .	49
5.24	Nodi halftoning con immagine texture . . . . .	50
5.25	Nodi halftoning con Voronoi texture . . . . .	51
5.26	Halftoning in coordinate UV . . . . .	51
5.27	Halftoning in coordinate camera . . . . .	51
5.28	Halftoning . . . . .	51
5.29	Nodi retino a righe con immagine texture . . . . .	52
5.30	Nodi retino a righe con Wave texture . . . . .	53
5.31	Retino a righe in coordinate UV . . . . .	53
5.32	Retino a righe in coordinate camera . . . . .	53
5.33	Retino a righe . . . . .	53
5.34	Ologramma . . . . .	54
5.35	Nodi shader ologramma . . . . .	55
5.36	Grafico dell'effetto noise sul valore strength dell'emission nel tempo	55
5.37	Effetto paint brush . . . . .	56
5.38	Unwrap UV della proiezione . . . . .	57
6.1	Temperatura colore . . . . .	59
6.2	Three point lighting . . . . .	61
6.3	Luci di blender . . . . .	62
6.4	Temperatura colore in Blender . . . . .	63
6.5	Reference . . . . .	64
6.6	Palette colori . . . . .	64
6.7	Luci ambiente(1) . . . . .	65

*Elenco delle figure*

---

6.8	Luci ambiente(2)	65
6.9	Luci ambiente(3)	66
6.10	Luci ambiente(4)	66
7.1		67
7.2		68
7.3		68
7.4		69