

FONDAMENTI DI COMPUTER GRAPHICS M (8 CFU)

LAB 06 - RENDERING CON BLENDER - DIGITAL ART

Michele Righi

Settembre 2023



Figura 1: Scena renderizzata con Cycles in Full HD

Traccia

L'obiettivo è quello di utilizzare il rendering di Blender per creare una scena 3D eventualmente quella del Lab 5 o sono ammessi anche oggetti presi da repository purché dichiarati. La scena deve mettere in evidenza le potenzialità di illuminazione, shading, materiali e texturing di Blender, confrontando la resa con Cycles (ray tracing-based) versus resa con Eevee (pipeline di rendering shaders-based). Si richiede una breve relazione in cui si descrivono le

tecniche utilizzate documentate da alcuni screenshot della scena resa. Generate la miglior immagine della vostra resa e allegatela alla relazione, l'immagine parteciperà al contest "Digital Art FCG2023"!! Si valuteranno le strategie di rendering utilizzate e gli effetti creati così come l'estetica di quanto prodotto sia questo terribilmente bello che accuratamente brutto, non si valuta la modellazione geometrica degli oggetti in scena. Buon divertimento!
NOTA: L'utilizzo di Blender non è vincolante: lo stesso esercizio puo' essere svolto in altri ambienti di modellazione/grafica che supportino illuminazione globale con ray tracing (es. POVRAY).

1 Introduzione

Per il progetto di questo laboratorio, ho voluto realizzare un soggiorno degli anni '60/'70, prendendo ispirazione dall'immagine mostrata in Fig. 2.



Figura 2: Famiglia in un salotto del 1960

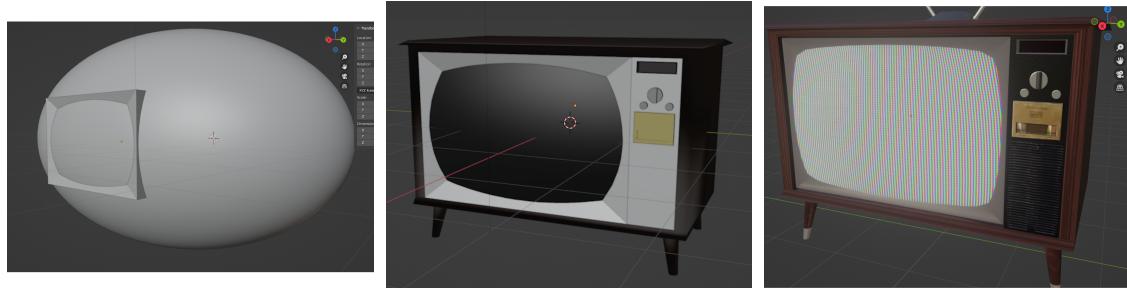
2 Implementazione

2.1 Televisore

Poiché volevo che l'elemento di scena principale fosse un televisore, rendendolo anche l'unica fonte di illuminazione, ho iniziato creandone un modello.

Per la realizzazione dello schermo ho utilizzato una sfera schiacciata che ho poi ritagliato per ottenere la superficie curva (Fig. 3a). Ho quindi modellato il contenitore esterno, i

controlli, l'esterno in legno, i piedi e l'antenna. Per renderlo più realistico, ho creato i materiali dei diversi elementi, tra cui legno (tramite UV mapping), plastica normale e ingiallita. Per lo schermo ho seguito *questo tutorial* che spiegava come realizzare un materiale per lo schermo di televisori a tubo catodico, ovvero costituito da pixel con i 3 diversi canali rosso, verde e blu (vedi Fig. 3c). Il risultato è visibile in Fig. 4.



(a) Modellazione schermo

(b) Modello completo

(c) Texturing e UV mapping



Figura 4: Televisore a tubo catodico

Poiché volevo che lo schermo illuminasse anche la stanza, ho indirizzato l'uscita del materiale creato verso uno shader tipo *Emission*, che ho combinato con un altro shader *Emission*, il quale riceve in ingresso l'uscita di un *Light Path*.

2.2 Tavolino

Come per il televisore, il modello del tavolo, assieme agli oggetti posizionati sopra di esso (eccetto il libro) sono stati modellati da me. Ho sfruttato in particolare le funzioni di *bevel*, *sheer*, *ridimensionamento*, ed i modificatori *subdivision surface*, *array* e *mirror*.

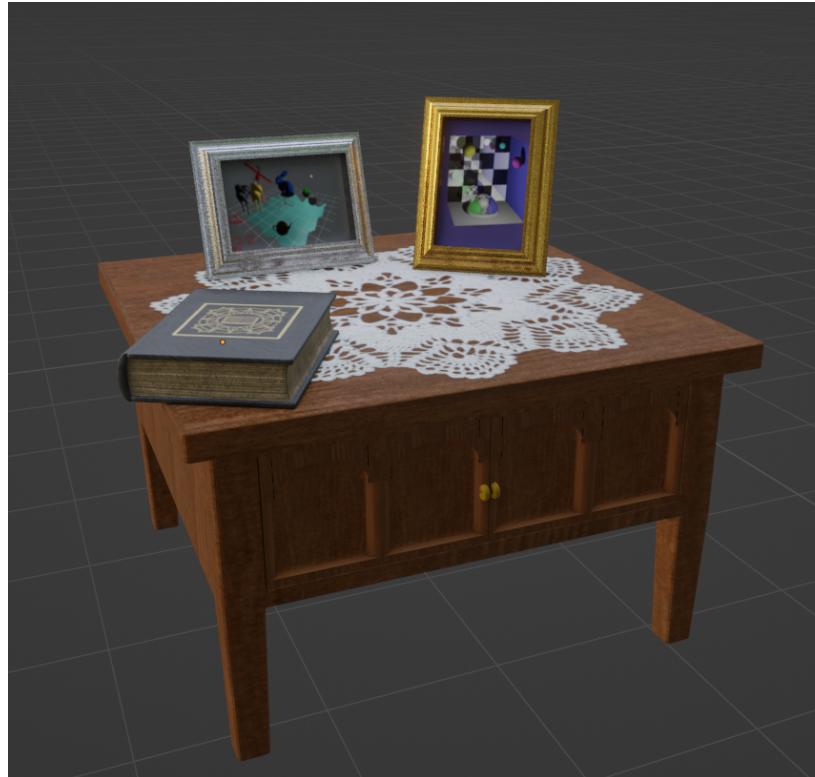


Figura 5: Tavolino

2.3 Stanza

Per realizzare la stanza ho utilizzato 4 piani: 2 piani verticali per le pareti, 2 piani orizzontali rispettivamente per pavimento e soffitto. Per le pareti ho utilizzato una texture di carta da parati e per il tetto il materiale di default. Per il pavimento ho seguito *questo tutorial* per creare un materiale che assomigliasse alla moquette.

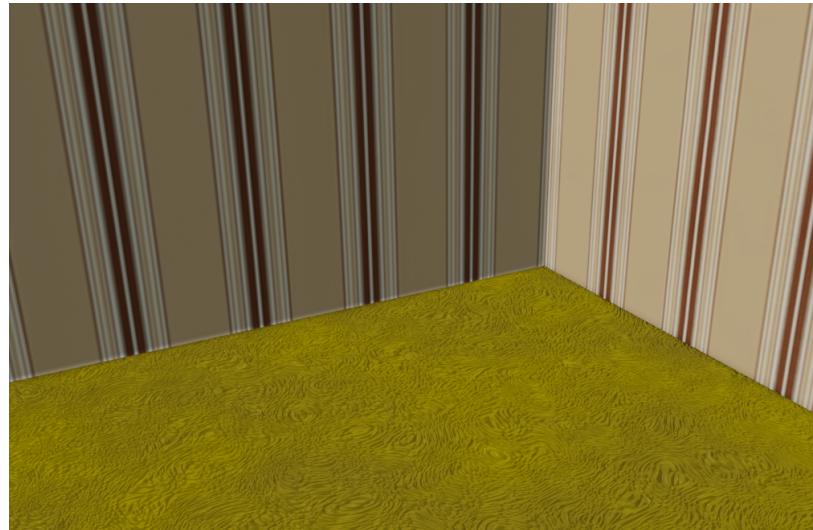


Figura 6: Stanza

2.4 Altri Oggetti di Scena

Per costruire la scena ho utilizzato diversi modelli presi dalle repository dei siti web più famosi:

- **Console Atari2600** - *Atari 2600* di Vaclav Marena;
- **Libro** - *Book* di James.Moore;
- **Divano** - *Old Plaid Sofa* di lena-wachs;
- **Lampada** - *old lamp lowpoly* di Renee Beenen;
- **Quadro** - *Fancy Picture Frame* di Jamie McFarlane, a cui ho modificato la texture per cambiare l'immagine;
- **Orologio a pendolo** - *Old Wooden Grandfather Clock* di Toni García Vilche;
- **Tende** - *Animated Curtains Low Poly* di JeffK, a cui ho modificato la struttura e le texture (legno e tessuto);
- **Pianta** - *Indoor Plant* di 3DHaupt;
- **Pianoforte verticale** - *Retro piano* di DailyArt;
- **Orsacchiotti di peluche** - *Teddy bears* di hectopod;
- **Cavallo a dondolo** - *Rocking Horse* di mervemacit;
- **Trenino giocattolo** - *Train Toy Wood* di Kazu_artts;

- **Macchina giocattolo** - *Toy Car_free model* di andree, a cui ho cambiato la texture;
- **Cubi giocattolo** - *Alphabet Blocks GameReady Free low-poly 3D model* di antonushakov94, a cui ho aggiunto materiali e colori, in quanto mancavano;

3 Rendering

3.1 Informazioni sul Dispositivo

Per il rendering delle scene ho utilizzato un computer portatile con le seguenti specifiche:

- Processore Intel® Core™ i7-7700HQ CPU @ 2.80GHz, 2808 Mhz, 4 core, 8 processori logici;
- RAM 16GB;
- Scheda grafica dedicata NVIDIA GeForce GTX 1060;
- Blender v3.4;

3.2 Cycles VS Eevee

Cycles ed Eevee sono i due motori di rendering di Blender. Si basano sulle due principali tecniche di rendering:

- **Cycles**, sfrutta il *ray-tracing* e utilizza un path tracer basato sulla fisica. Fornisce un rendering generalmente più accurato ma impiega molto tempo (offline rendering);
- **Eevee**, sfrutta il PBR (*Pipeline-Based Rendering*), ovvero comprende la fase di rasterrizzazione e utilizza OpenGL. Fornisce un rendering generalmente meno accurato della controparte, ma molto veloce, che permette la rappresentazione di scene in real-time e l'interattività.

3.3 Illuminazione in Eevee

Provando a renderizzare la scena sfruttando l'illuminazione generata del televisore in Eevee, il risultato è quello mostrato in Fig. 7.



Figura 7: Rendering senza bake lighting

Questo succede perché in Eevee, ogni luce che arriva da un oggetto che non è una fonte luminosa vera e propria, di base non viene considerata [1]. Ciò si applica ad esempio all'illuminazione data dal mondo (World) e alle mesh a cui sono associati nodi di tipo Emission (dunque anche lo schermo del televisore, in quanto è una sorta di effetto glow dato dalle proprietà del suo materiale).

Per ottenere un calcolo delle luci accurato su Eevee, nel nostro caso è necessario aggiungere un **Irradiance Volume** ed effettuare il *bake indirect lighting*, prima del rendering.



Figura 8: Rendering in Eevee con bake lighting e risoluzione a 10 campioni su ciascun asse, Full HD. Tempo impiegato per il bake: 2 minuti



Figura 9: Rendering in Eevee con bake lighting e risoluzione a 25 campioni su ciascun asse, Full HD. Tempo impiegato per il bake: 34 minuti



Figura 10: Rendering in Cycles, Full HD. Tempo impiegato per il rendering: 3 minuti e mezzo

3.4 Altre Immagini

Ho eseguito diversi rendering, i risultati dei quali sono, tuttavia, dei file di dimensioni molto grandi. Dunque di seguito riporto dei link per poterli scaricare dalla mia repository GitHub:

<i>Rendering Engine</i>	<i>Risoluzione</i>	<i>Tempo di Rendering</i>	<i>Download Links</i>
Cycles	Full HD (1920x1080)	3"30'	<u>8bit colors</u>
Cycles	4K (3840x2160)	14"21'	<u>8bit colors</u> <u>16bit colors</u>
Cycles	8K (7680x4320)	42"15'	<u>8bit colors</u> <u>16bit colors</u>
Eevee	Full HD (1920x1080)	4'	<u>8bit colors</u>
Eevee	4K (3840x2160)	12'	<u>8bit colors</u> <u>16bit colors</u>
Eevee	8K (7680x4320)	51'	<u>8bit colors</u> <u>16bit colors</u>

Riferimenti bibliografici

[1] Blender. Indirect lighting in eevee.