

CAPITOLO 3

Il Nodo Mix del gruppo Color

Ora che abbiamo preso in esame i **Nodi** di input e di output, possiamo iniziare la trattazione dei **Nodi** intermedi, o “**filtri**”, che consentono l’**elaborazione** delle informazioni e, quindi, il **Post-Processing** vero e proprio.

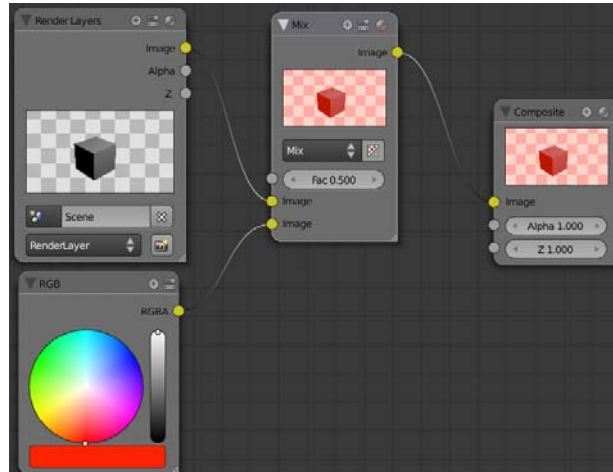
Iniziamo con il **Nodo Mix**, presente all’interno del gruppo **Color**, che mette a disposizione diverse funzioni per “**miscelare**”, come suggerisce il nome, due informazioni diverse in input per produrre un’unica **immagine in output**; si tratta, in effetti, di uno dei **Nodi** utilizzati con maggiore frequenza in fase di **Compositing**.

Per aggiungere un **Nodo Mix** al **Node Editor** è sufficiente cliccare su **Add → Color → Mix**.

Aggiungiamo anche un altro **Nodo** di input, oltre a **Render Layers**, ad esempio **RGB** con colore **rosso** e Alpha a 1, in modo da provare gli **effetti** di miscelazione di **Mix**.

Colleghiamo l’**output** di **Mix** all’input **Image** di **Composite**.

Come visibile nella **figura** in basso, con queste impostazioni (**Mix** è la modalità di miscelazione di default, **Factor** 0.5 è il fattore di miscelazione predefinito) aggiungeremo una **tinta rossa uniforme** a tutta l’immagine finale da renderizzare, in quanto stiamo miscelando i due elementi con un valore di **mixaggio** pari a **0.5**; portando **Factor** a 0 o 5, varieremo l’entità della tinta rossa sullo sfondo rappresentato dall’output del **Render Layer**.

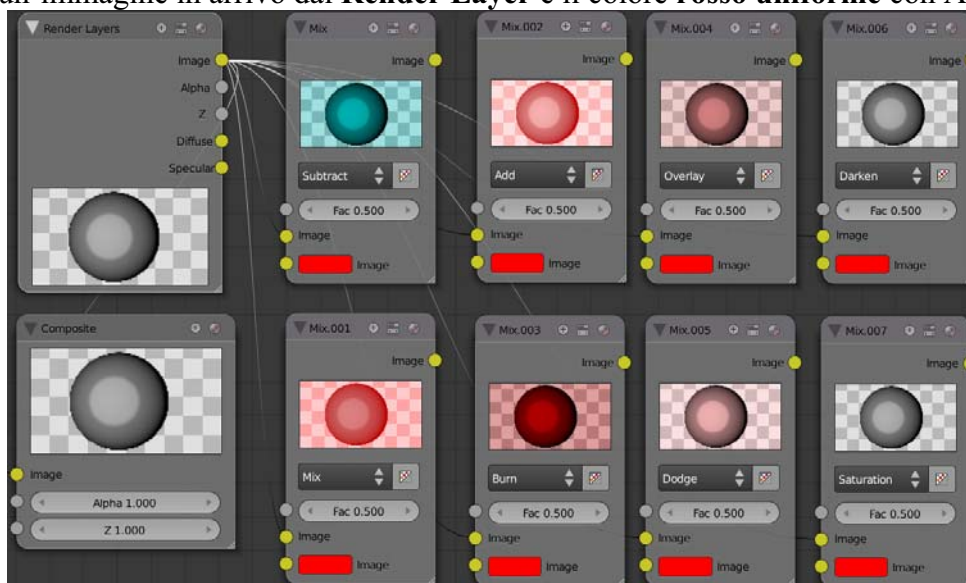


Aprendo il menù a tendina del **tipo di filtro**, in **Mix**, possiamo scegliere tra le diverse **modalità di miscelazione** messe a disposizione da **Blender 2.5**:

- **Mix** --- La prima immagine viene coperta, pixel per pixel, dalla seconda, in base al Factor e ai valori della trasparenza (Alpha).
- **Hue** --- I valori dei pixel vengono considerati nello spazio colore HSV; in questo caso, i valori di Tonalità (Hue) dei due pixel vengono mescolati, il risultato viene combinato con Intensità (Value) e Saturazione del primo pixel e riconvertito nello spazio dei colori RGB.
- **Saturation** --- I valori dei pixel vengono considerati nello spazio colore HSV; in questo caso, i valori di Saturazione dei due pixel vengono mescolati, il risultato viene combinato con Intensità (Value) e Tonalità (Hue) del primo pixel e riconvertito nello spazio dei colori RGB.

- **Value** --- I valori dei pixel vengono considerati nello spazio colore HSV; in questo caso, i valori di Intensità (Value) dei due pixel vengono mescolati, il risultato viene combinato con Tonalità (Hue) e Saturazione del primo pixel e riconvertito nello spazio dei colori RGB.
- **Color** --- Aggiunge un colore ad un pixel, in modo da fornire una determinata tinta a tutta l'immagine.
- **Burn** --- Scurisce la prima immagine basandosi sul gradiente fornito con la seconda immagine di input; il risultato generale è, quindi, un'immagine più scura.
- **Dodge** --- Schiarisce la prima immagine basandosi sul gradiente fornito con la seconda immagine di input; il risultato generale è, quindi, un'immagine più chiara.
- **Linear Light** --- Una combinazione di Burn e Dodge, a seconda del colore di base.
- **Lighten** --- Si fa un confronto pixel per pixel e si prende, dei due, quello più chiaro, applicandolo all'immagine finale.
- **Darken** --- Si fa un confronto pixel per pixel e si prende, dei due, quello più scuro, applicandolo all'immagine finale.
- **Difference** --- Da non confondere con Subtract, in quanto qui viene preso il valore assoluto della differenza, per cui il nero indica colori uguali, bianco colore opposto e i valori intermedi sono una "misura della distanza" pixel per pixel tra le due immagini.
- **Divide** --- Divide la prima immagine per la seconda, pixel per pixel e canale per canale; in caso di secondo pixel a 0.0 (nero), per evitare la divisione per 0 (impossibile) Blender lascerà il valore del pixel della prima immagine.
- **Overlay** --- Una combinazione di Screen e Multiply, a seconda del colore di base.
- **Screen** --- I valori dei due pixel vengono invertiti, moltiplicati tra di loro e il risultato viene a sua volta invertito.
- **Multiply** --- Moltiplica la seconda immagine, pixel per pixel e canale per canale, alla prima, "pesando" l'effetto con il controllo Factor. Maggiori dettagli sotto.
- **Subtract** --- Sottrae la seconda immagine, pixel per pixel e canale per canale, alla prima, "pesando" l'effetto con il controllo Factor. Maggiori dettagli sotto.
- **Add** --- Somma la seconda immagine, pixel per pixel e canale per canale, alla prima, "pesando" l'effetto con il controllo Factor. Maggiori dettagli sotto.

L'immagine seguente mostra i **risultati** ottenuti con diversi filtri **Mix** con **Factor** a 0.5 utilizzando un'immagine in arrivo dal **Render Layer** e il colore **rosso uniforme** con Alpha a 1.



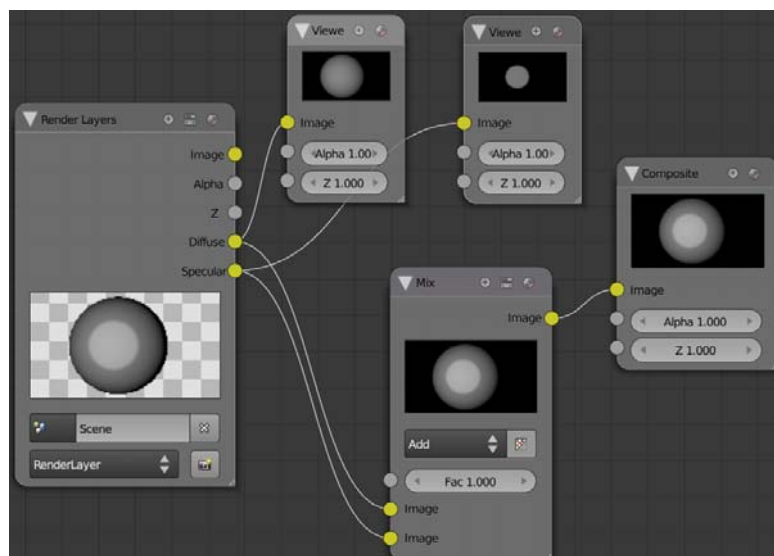
Spendiamo ora qualche parola per definire come operano i **filtri "aritmetici"** come **Divide**, **Multiply**, **Subtract** e **Add**.

Come è noto, ciascun **pixel** porta con sé le informazioni sul **colore** memorizzando quattro valori: **R** (rosso), **G** (verde), **B** (blu) e **A** (Alpha, la trasparenza); il **range di valori** va da 0.0 (nullo) a 1.0 (totale).

I **filtri** lavorano “**per pixel**” e “**per canale**”, ossia modificando il valore di ciascun **pixel** e, per ciascun **pixel**, prendendo i vari **canali** singolarmente, durante il processo di **elaborazione**; nel caso dei **filtri aritmetici**, vengono effettuate proprio le **operazioni** di moltiplicazione, sottrazione e addizione dei valori, con alcune conseguenze:

- **moltiplicare** due valori minori di 1, come nel nostro caso, porta ad un risultato che sarà 1 solo con due 1 in input (infatti, ad esempio, $0.5 \times 0.5 = 0.25$, mentre $0.25 \times 1.0 = 0.25$), per cui in generale l'effetto di Multiply sarà quello di scurire il pixel e, quindi, l'immagine in output; moltiplicare infine un qualsiasi valore per 0.0 (pixel nero) produrrà in output un pixel nero;
- **sommare** due valori con Add produrrà sempre un pixel più chiaro dei due originali (a meno che non siano entrambi a 0.0); il valore massimo è 1.0, per cui qualora la somma dovesse superare tale limite (es.: $0.5 + 1.0$), Blender imporrà come valore finale 1.0;
- **sottrarre** un valore ad un altro con Subtract porta, chiaramente, ad un valore più basso, quindi a pixel e immagini più scure; vale la considerazione del limite fatta per Add, ma verso il basso: valori minori di 0.0 (es.: $0.5 - 1.0$) verranno comunque posti a 0.0 da Blender;
- **dividere** un numero minore di 1 per un altro può portare sia a valori minori di 1 (es.: $0.6 / 0.8 = 0.75$) che maggiori di 1 (es.: $0.5 / 0.2 = 2.5$), ma in ogni caso valgono le regole di troncamento dei valori a 1.0 viste per Add.

Vediamo un **esempio** pratico di **addizione** di valori miscelando tra loro i canali **Diffuse** e **Specular** di un'immagine, impostando il **Node Editor** come visibile in **figura** sotto, in modo da ottenere un **rendering** fatto esclusivamente da questi due **canali** (per accentuare l'effetto del riflesso **Speculare**, ho utilizzato un ombreggiatore **Toon** per **Specular**).

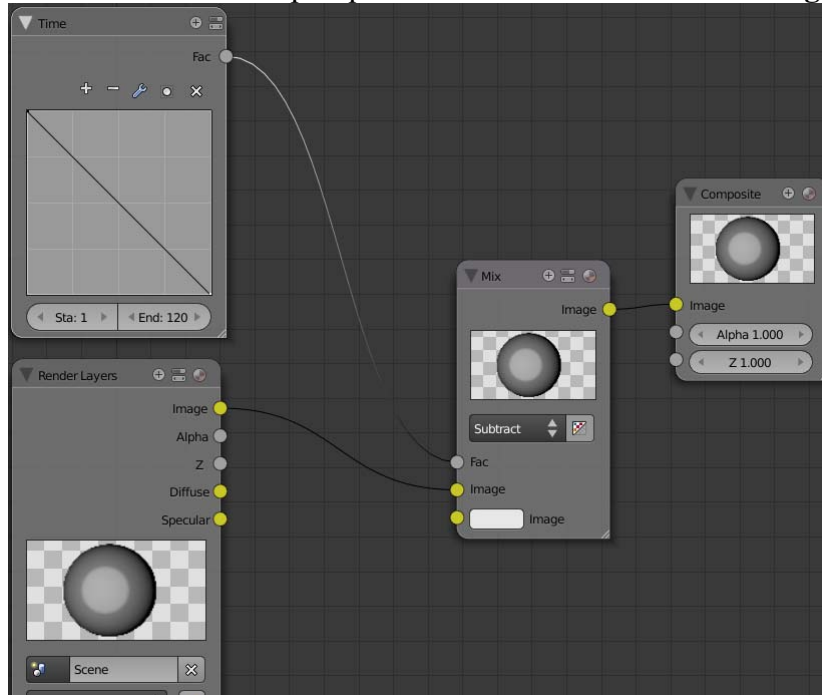


Nel **Node Editor** ho inserito anche due **Nodi Viewer**, collegandoli a **Diffuse** e **Specular** di **Render Layers**, in modo da valutare singolarmente i contributi di informazione dati dai due **canali**.

Torniamo alla trattazione generale del **Nodo Mix**.

Come avrete notato, anche per **Factor** è presente una **porta di input**: ciò significa che il **valore di miscelazione** non deve essere per forza **costante** per tutta l'immagine o per tutta l'animazione, ma può **variare**, ad esempio, in base ad una **mappa**, una **ColorBand** o al numero di **frame** corrente; vediamo subito come far variare **Factor** da 0 a 1 nell'arco di **120 frames** per rendere **più chiara** l'immagine, come in una **dissolvenza** in entrata.

La configurazione del **Node Editor** per questo **effetto** è mostrata nell'**immagine** seguente.



Utilizziamo un **Nodo** di tipo **Input – Time** con la **curva invertita** e colleghiamo alla porta di input **Factor** di **Mix**.

Colleghiamo l'uscita **Image** del **Render Layer** alla prima porta **Image** di **Mix**, lasciando la **seconda porta vuota** ed impostando per la stessa il **colore bianco** (in assenza di immagini in input, quindi, è possibile impostare un colore uniforme per i canali in input di **Mix**).

Cambiamo il **tipo di operazione** in **Subtract**: in questo modo, al frame 1 **sottrarremo** alla prima immagine (in arrivo dal **Render Layer**) il **colore bianco uniforme**, per cui in pratica la azzereremo (sottraendo 1.0 ad ogni pixel) ed avremo un'immagine nera, ma andando avanti con i **frame** il **fattore** di miscelazione diminuirà seguendo l'andamento della curva definita in **Time**, per cui **alla fine** (al frame 120, nel nostro caso) **la sottrazione sarà nulla** e avremo l'immagine del **Render Layer** in output.

L'**immagine** qui sotto mostra il **risultato** visibile in **Composite** ai frame 1, 60, 90 e 120.



Per **accelerare** o **rallentare** la **dissolvenza**, è sufficiente **modificare la forma della curva in Time**, inserendo e spostando dei punti di controllo con un click del tasto sinistro del mouse sulla **curva**.

* * *