

Blockink

TUTORIAL

Copyright: cameyo 2018

License: MIT

Versione BlackLink: 0.357

INDICE

Introduzione	5
Panoramica	7
Area di Lavoro	9
Pennelli.....	10
Anteprima Pennello	13
Storia delle impostazioni del pennello corrente.....	14
Anteprima interattiva del pennello	15
Colori	17
Accesso rapido ai colori	17
Cronologia dei colori (Storia del colore)	17
Colori preferiti.....	19
Tipi di Colore	20
Colore a tinta unita	21
Colore sfumato (gradiente)	24
Colore da immagine.....	29
Cursori (Slider)	34
Segmenti (Segments).....	35
Forma del pennello (Shapes).....	37
Tutorial: Primi passi	39
Tutorial: Navigazione e Visualizzazione (Navigate & View).....	50
Tutorial: Selezione e Trasformazione (Select & Transform).....	53
Tutorial: Riempimento (Fill)	59
Tutorial: Modalità di Disegno (Drawing Mode).....	60
Tutorial: Mostra Finestre/Pannelli (Show)	63
Tutorial: Zoom dell'Interfaccia Grafica	65
Tutorial: Sostituzione dei pennelli predefiniti	66
Tutorial: Modifica veloce della dimensione, dell'opacità e della morbidezza del pennello.....	68
Tutorial: Cambio veloce tra Pennello e Gomma (Eraser)	70
Tutorial: Cattura dei colori dal disegno.....	70
Tutorial: Colori preferiti.....	71
Tutorial: I Cursori (Slider)	72
Tutorial: I Livelli.....	73
Tutorial: I Gradienti	76
Tutorial: La finestra secondaria (Secondary View)	82
Tutorial: Creazione di un disegno da un'immagine (color ImageBased)	85
Tutorial: La struttura dei file dei Colori Preferiti	90

Tutorial: Le cartelle utente di BlackLink	93
Brush workshop 0: Introduzione ai controller	98
Brush workshop 1: Pennello quadrato	104
Brush workshop 2: La funzione Seno	113
Brush workshop 3: Gradienti e controller.....	122
Brush workshop 4: Matematica e controller	131
Brush workshop 5: Esperimenti con un pennello base	137
Pennello che modifica la Dimensione (Size) in maniera casuale durante la pennellata.....	143
Pennello che disegna gruppi di punti sparsi di dimensione casuale.....	144
Pennello che crea pennellate tratteggiate.....	148
Brush workshop 6: Il Pennello Tubo	153
Breve introduzione al Brush Shading Language (BSL)	156
Utilizzo del Brush Shader Editor	157
Analisi di uno shader (hardness ctrl).....	162
La funzione main e la struttura idatas	164
Esempi di BSL	166
Tutorial: Impostazione della tavoletta Wacom con BlackLink	197
Controller Reference.....	205
Global Parameters	205
Stroke Parameters	206
Math F(X).....	208
Operators	210
Cast Operators	212
Signals.....	213
Tasti scorciatoia (Shortcuts)	215
Shortcuts BlackLink	217
Pennelli predefiniti (ordine alfabetico).....	221

Introduzione

Questa è una raccolta di tutorial sul programma **BlackInk by Bleank**, un programma per disegnare orientato verso la produzione di Generative Art.

Alcuni di essi sono stati presi dal forum (<http://forum.bleank.com/>) e dalla documentazione del programma (<http://blackink.bleank.com/Documentation/>), mentre altri sono originali.

I tutorial non originali sono stati espansi, modificati e aggiornati alla versione 0.357 del programma.

Questo è un libro in continuo aggiornamento... potete trovare l'ultima versione all'indirizzo web:

<https://github.com/cameyo42/Black-Ink-Tutorial>

Troverete che alcune informazioni sono ripetute più volte... repetita iuvant.

Tutti gli errori che incontrerete sono dovuti alla mia inesperienza o alla mia errata comprensione.

Sistema:

- BlackInk 0.357
- Windows 10 Professional
- RAM 16GB
- GeForce GTX 750
- Wacom Intuos Draw (medium)

NOTA:

Data la rapida evoluzione del programma, le immagini e i risultati delle operazioni potrebbero essere diversi.

Un altro fattore che differenzia i risultati è la modalità di input, cioè se usiamo una Tavoletta oppure un Mouse (inoltre ogni tavoletta si comporta in maniera diversa nei confronti della Pressione).

Lo scopo dei tutorial è quello di spiegare le funzionalità del sistema e permettere agli utenti di sperimentare per conto proprio.

NOTA:

Sarete in grado di sfruttare tutte le potenzialità di BlackInk potenzialità solo con tavolette grafiche sensibili alla pressione e all'inclinazione (Tilt).

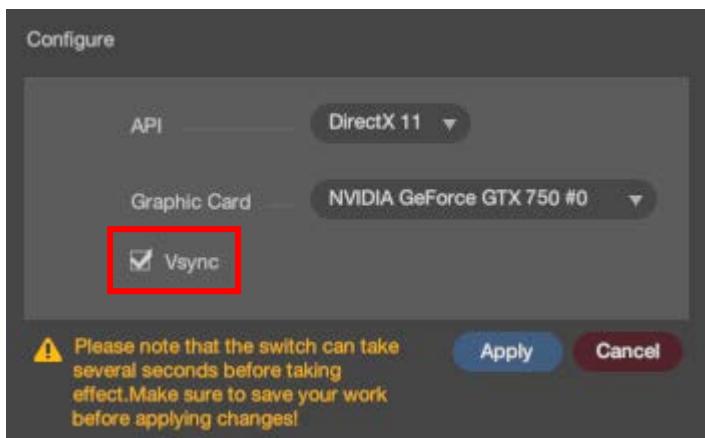
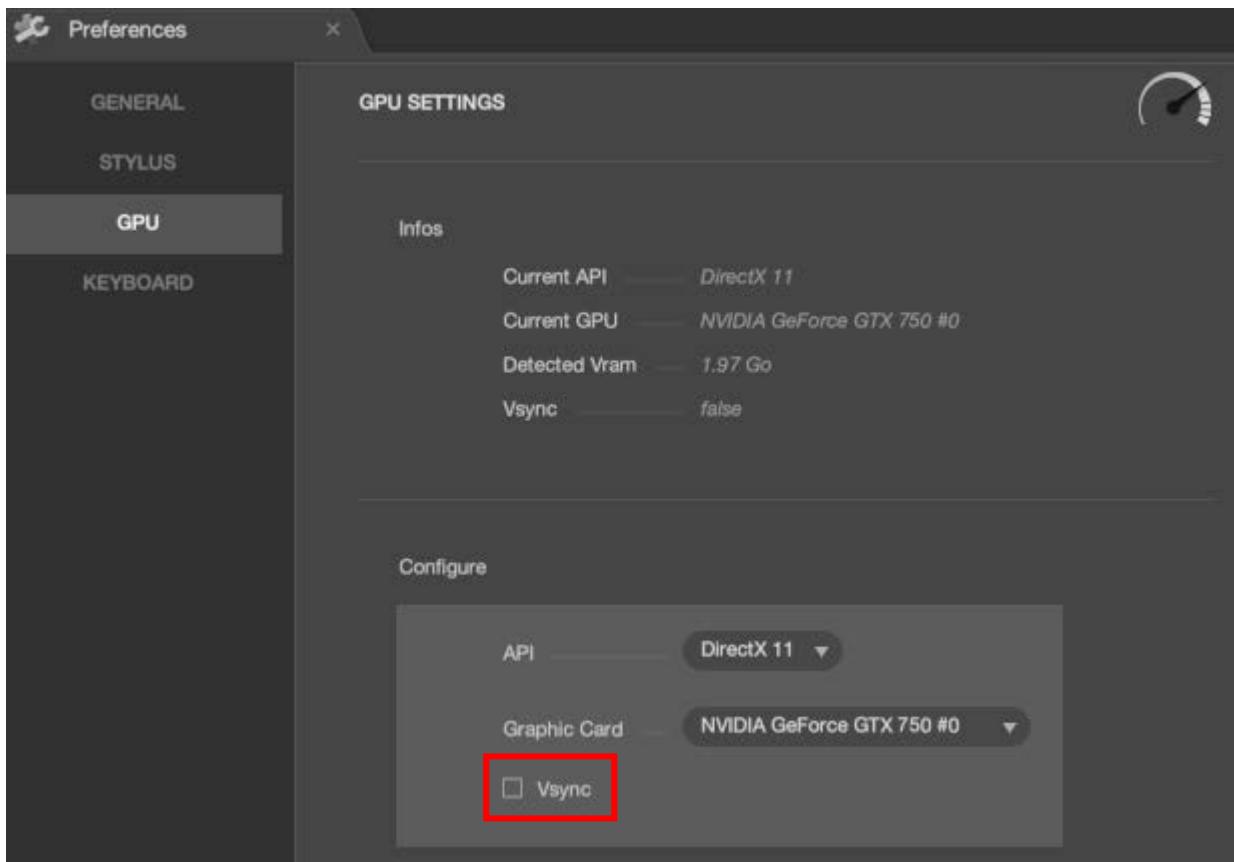
NOTA:

BlackInk è un programma che consuma molte risorse del computer (RAM, CPU e soprattutto GPU):

Name	CPU	Memory	Disk	Network	GPU
Apps (5)					
>  BlackInk.exe (32 bit)	31.3%	602.4 MB	0 MB/s	0 Mbps	98.6%

Quindi vi consiglio di chiudere tutte le altre applicazioni mentre disegnate.

Potete provare a diminuire il consumo di GPU abilitando il parametro **VSync** che si trova nella sezione **Preferences -> GPU** del menu **Window->Preferences**:



Inoltre la scheda grafica dovrebbe avere un valore di PassMark G3D di almeno 2000; potete trovare il valore della vostra scheda al seguente indirizzo: https://www.videocardbenchmark.net/gpu_list.php

Panoramica

Questa panoramica segue la documentazione fornita dal programma (che non è aggiornatissima) ed è una presentazione delle caratteristiche e funzionalità dell'applicazione. Non è indispensabile per seguire i tutorial successivi, ma permette rapidamente di farsi un'idea sulle potenzialità del programma ed è fortemente consigliato.

Perfettamente reattivo

BlackInk è un software di pittura digitale che utilizza l'hardware grafico del tuo computer per fornire un'esperienza perfettamente reattiva indipendentemente dalle dimensioni del tuo disegno.



Generative Art

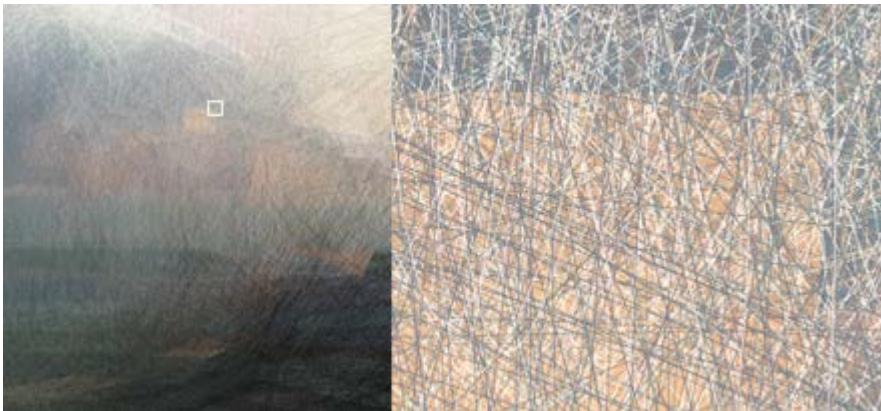
BlackInk non copia altre soluzioni software, crea nuovi strumenti grafici.

Si presenta come un software di "arte generativa" perché non tenta di imitare l'acquerello o la pittura a olio, sfrutta tutto il potenziale del tuo computer per creare nuove possibilità come gradienti di tratto e generazione del colore da immagini fornite dall'utente.



Risoluzioni molto alte

La reattività è la sua più grande risorsa. La pittura è sempre immediata e reattiva, senza alcun ritardo generato dal computer anche con risoluzioni molto elevate.

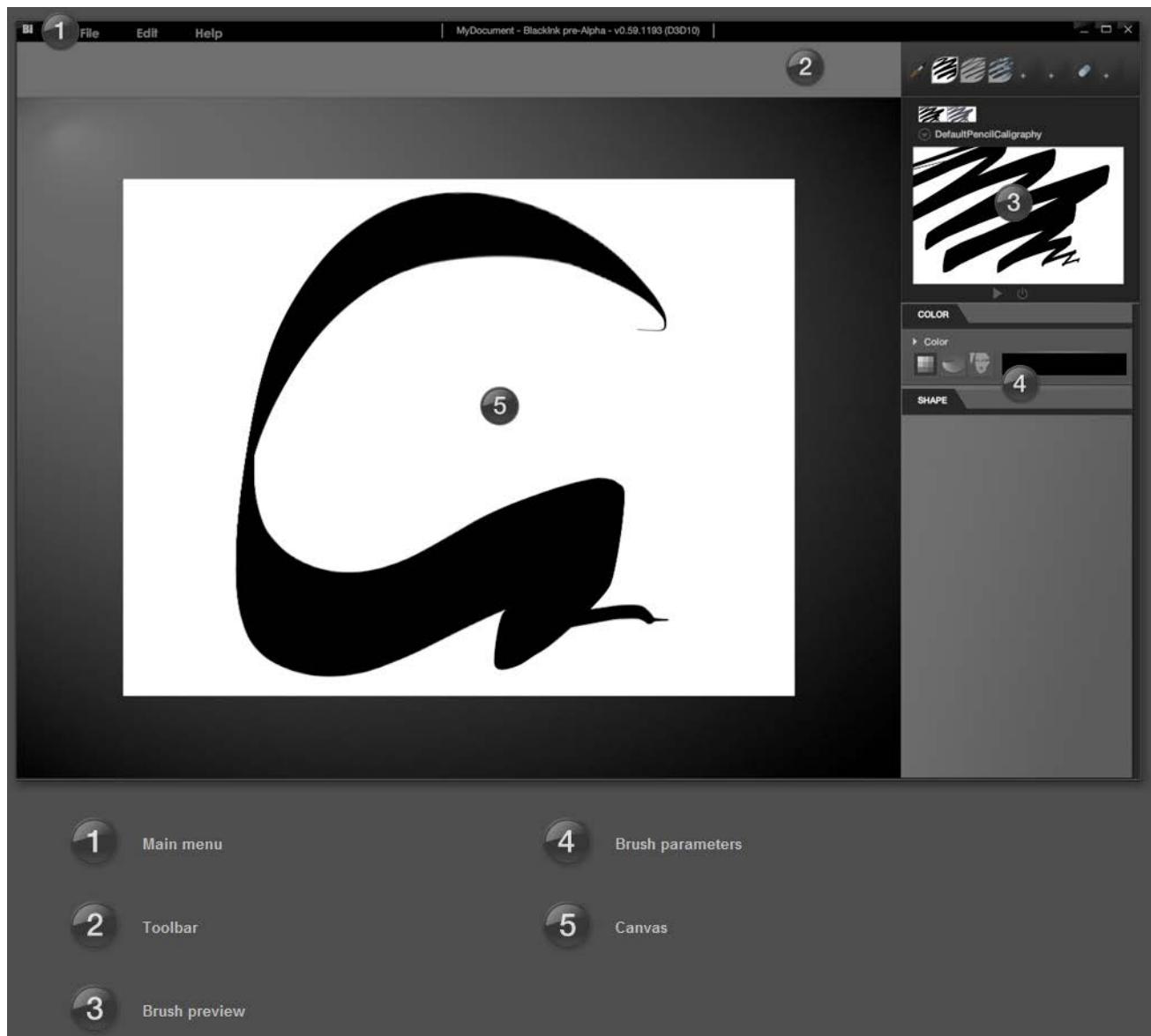


Interfaccia user-friendly

BlackInk ha anche un approccio diverso all'esperienza utente: accesso immediato ai parametri del pennello con cronologia delle impostazioni e cronologia dei colori, anteprima istantanea e personalizzazione dei pennelli con finestre delle impostazioni in tempo reale e molte altre innovazioni user-friendly.

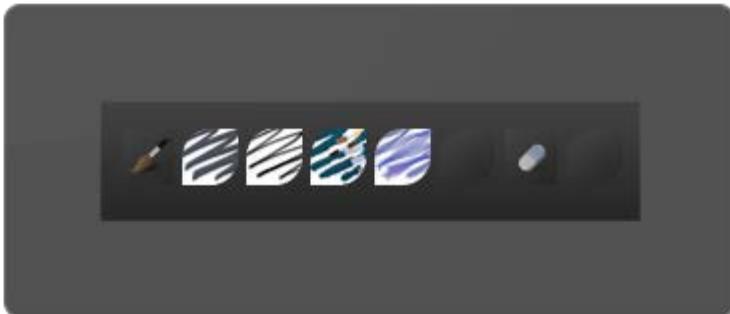


Area di Lavoro



Pennelli

Questi sono i pennelli predefiniti:



Penna Stilografica - Calligraphic Pencil (pennello standard)



L' opacità aumenta con la pressione.

Lo spessore, la scala e la rotazione aumentano con la velocità della pennellata.

Pennarello - Felt Pen (pennello standard)



Lo spessore aumenta con la pressione

La sfocatura del pennello aumenta con la velocità della pennellata

Crepe (pennello a doppio colore)



Alternare rettangoli colorati.

Il colore prelevato dal gradiente e lo stiramento dei rettangoli cambia con la velocità del tratto.

La dimensione e l'opacità dei rettangoli aumenta con la pressione.

Lo spessore e la lunghezza dei rettangoli sono modificati casualmente in base al valore del parametro "casuale" (random).

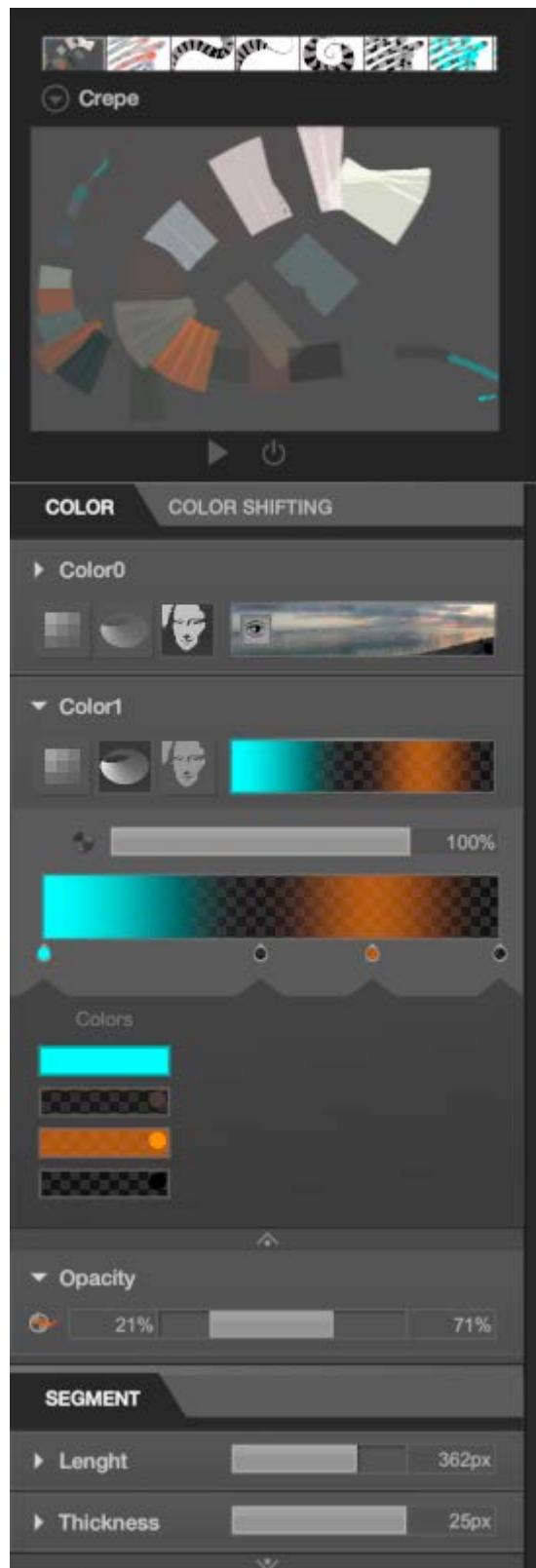
Volute (pennello con cavità)



Le dimensioni e la rotazione della forma cambiano con la velocità della pennellata.
Il colore selezionato dal gradiente e l'opacità del tratto cambiano con la pressione del tratto.

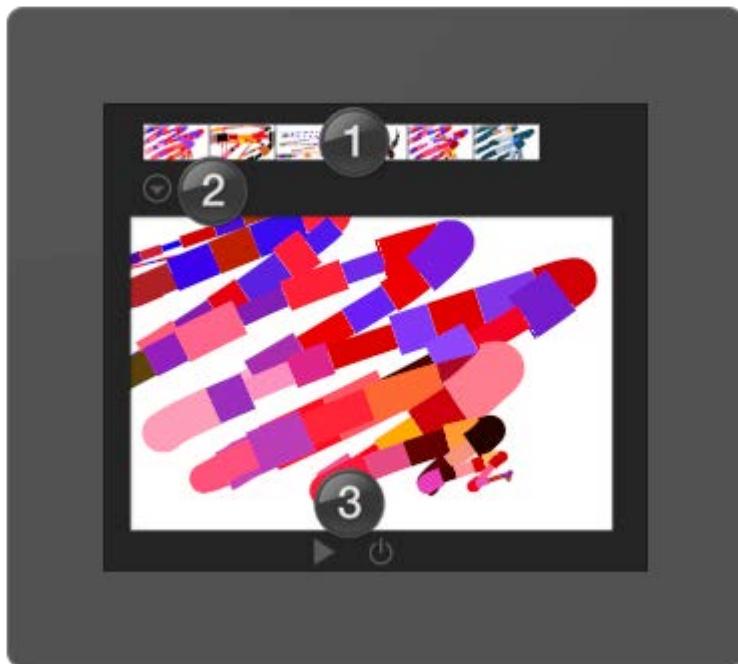
ALCUNI ESEMPI DI PARAMETRI DEI PENNELLI





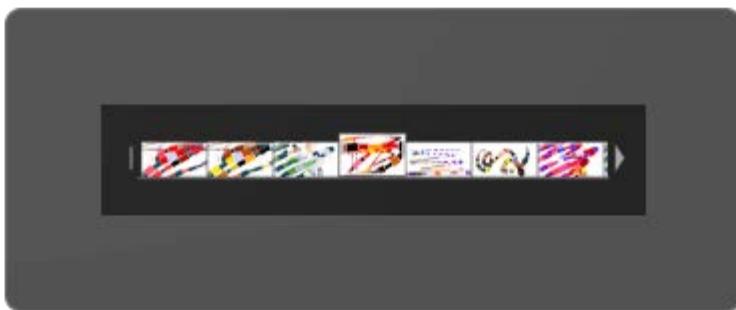
Anteprima Pennello

Le finestre Anteprima pennello riflettono le modifiche apportate ai parametri del pennello in tempo reale. Troviamo anche una cronologia del pennello per richiamare istantaneamente le impostazioni precedenti.



- 1** Storia delle impostazioni del pennello corrente
- 2** Menu del pennello
- 3** Anteprima interattiva del pennello e relativo player

Storia delle impostazioni del pennello corrente

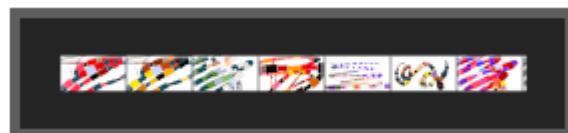


La cronologia attuale del pennello viene sempre visualizzata sopra l'anteprima del pennello

Cronologia vuota: solo il pennello corrente con le sue impostazioni predefinite è visibile.



Cronologia piena: sono disponibili tutte le precedenti impostazioni per il pennello corrente.



Per scegliere un'impostazione precedente, è sufficiente fare clic su di essa.



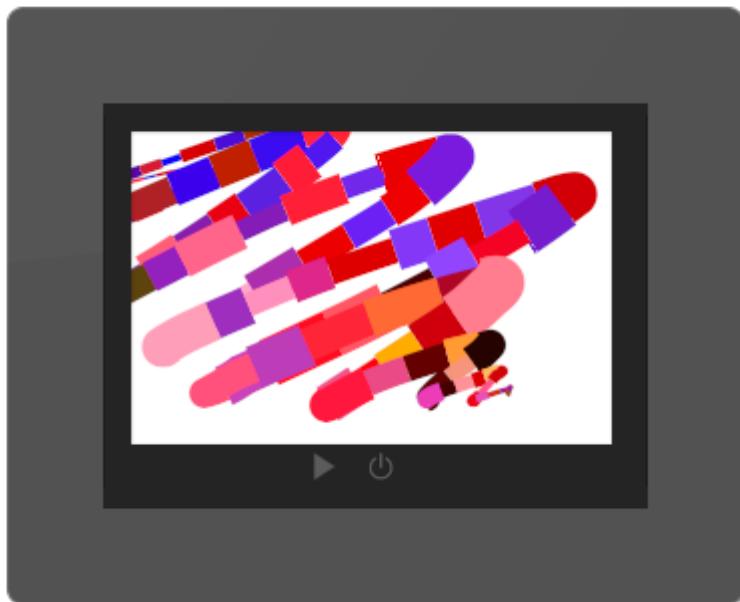
Per visualizzare le impostazioni precedenti, basta scorrere la freccia a destra o fare clic su di essa.



Per visualizzare le nuove impostazioni, basta scorrere la freccia a sinistra o fare clic su di essa.



Anteprima interattiva del pennello



Disegno predefinito utilizzato per visualizzare i pennelli.



Se passiamo con il mouse sopra l'anteprima, viene visualizzata un (gradiente dal bianco al nero) sul lato destro della zona di disegno.



Le modifiche apportate all'anteprima del pennello si riflettono nella cronologia del pennello: Quando si sceglie un valore nella barra della sfumatura, viene influenzato il colore di sfondo dell'anteprima del pennello.



Icona pennello sulla cronologia



È anche possibile sostituire il tratto di anteprima predefinito disegnando direttamente nell'area di anteprima.

Tutte le anteprime verranno aggiornate con il tuo ultimo tratto.



Icona pennello sulla cronologia

Il player dell'anteprima pennello

Questo player viene usato per visualizzare la dinamica della pennellata:



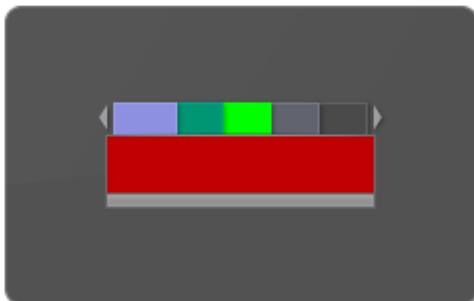
Riproduce l'animazione di anteprima del pennello



Ripristina il disegno dell'anteprima del pennello

Colori

Accesso rapido ai colori



Cronologia dei Colori

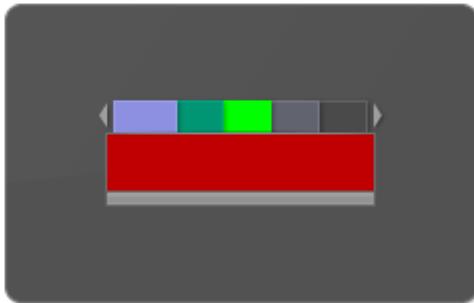
Visualizza le scelte di colore precedenti per l'accesso rapido e consente di cambiare l'opacità.



Colori preferiti

Qui puoi salvare e richiamare in modo permanente le tue impostazioni di colore preferite.

Cronologia dei colori (Storia del colore)



Ogni tipo di colore ha una propria cronologia per un facile accesso a ogni impostazione.

Per visualizzare la storia dei colori:



Rimani con il mouse sopra l'anteprima del colore.



Compare la cronologia dei colori sopra l'anteprima del colore.

Dettagli sulla storia dei colori:

A) Lista Colori

La parte superiore mostra la cronologia dei colori utilizzati in precedenza.

**B) Anteprima Colore**

La parte centrale è l'anteprima del colore corrente.

Quando fai clic su di esso si aprono le finestre dei parametri di colore.

**C) Cursore Alpha**

Nella parte inferiore il cursore rende il colore più o meno trasparente regolando il valore alfa.

**A) Lista Colori (per una scelta facile dei colori)**

Se si passa sopra un colore, questo si evidenzia.



Un clic per selezionare il colore. Il colore selezionato andrà nella prima posizione.



Se hai il cursore su una freccia, l'elenco dei colori scorrerà nella direzione della freccia.

La freccia a sinistra mostra i colori più recenti e la freccia a destra mostra i colori più vecchi.



Se fai clic sulle frecce scorrerai le icone dei colori una per una.



Se il colore è troppo trasparente per essere visto chiaramente nella miniatura, appare un indicatore che mostra il colore completo.

B) Anteprima Colore (Un modo semplice per visualizzare e modificare i colori)

Se fai clic sull'anteprima del colore, puoi modificarlo.

A volte le opzioni possono essere visualizzate in sovrapposizione:



Sul colore basato su immagine, viene visualizzato un occhio. Permette di mostrare o nascondere l'immagine di riferimento sulla tela.



Se il colore è troppo trasparente per essere chiaramente visibile nell'anteprima del colore, appare un indicatore che mostra il colore completo.



Nella tavolozza dei colori, lo sfondo rappresenta il nuovo colore e il cerchio rappresenta il colore corrente. Per tornare al colore corrente puoi fare clic sul cerchio.

C) Cursore Alpha



Quando si tiene premuto il pulsante sinistro del mouse e si sposta il cursore viene visualizzato il valore dell'alfa.



Non confondere "color alpha" con "brush opacity" che definisce i comportamenti del pennello.

Colori preferiti



Puoi sempre avere accesso ai colori che hai definito (colori solidi, gradienti (sfumati) e basati su immagini).

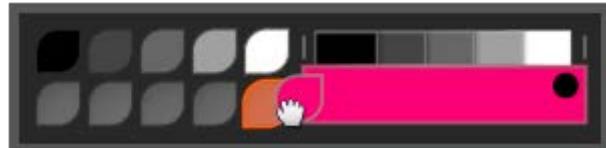
Avvio di BlackLink

I colori impostati come preferiti verranno automaticamente caricati all'avvio dell'applicazione.



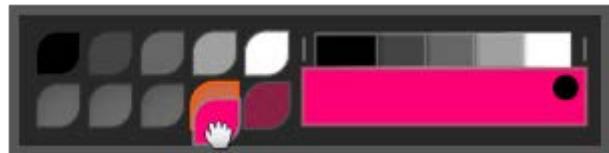
Salvataggio

Trascina e rilascia il colore desiderato su uno slot vuoto per salvarlo.



Copia

Per copiare un colore, trascinalo e rilascia un altro slot.



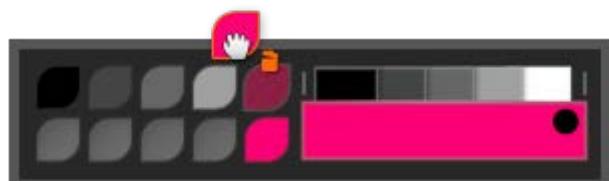
Sostituzione

Trascina e rilascia il tuo colore in un colore esistente.

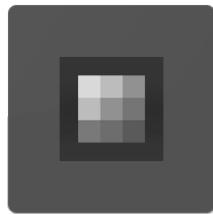


Eliminazione

Per eliminare il tuo colore, prendilo e trascinalo fuori dalla finestra dei preferiti e quando appare l'icona del cestino, rilascialo.



Tipi di Colore



Colore a tinta unita

Colore classico uniforme semplice.



Colore Sfumato

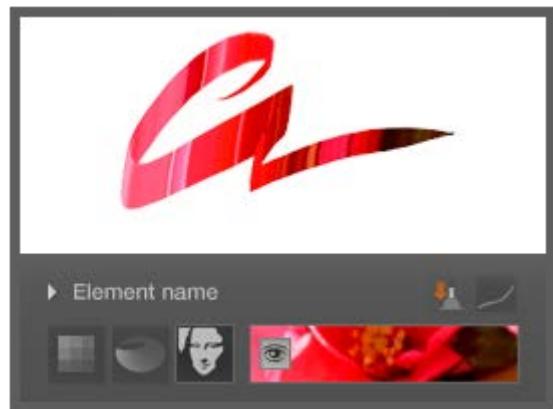
Il pennello ha un comportamento che segue l'impostazione del gradiente.





Colore da immagine

Il colore viene selezionato automaticamente da un'immagine fornita dall'utente.



Colore a tinta unita

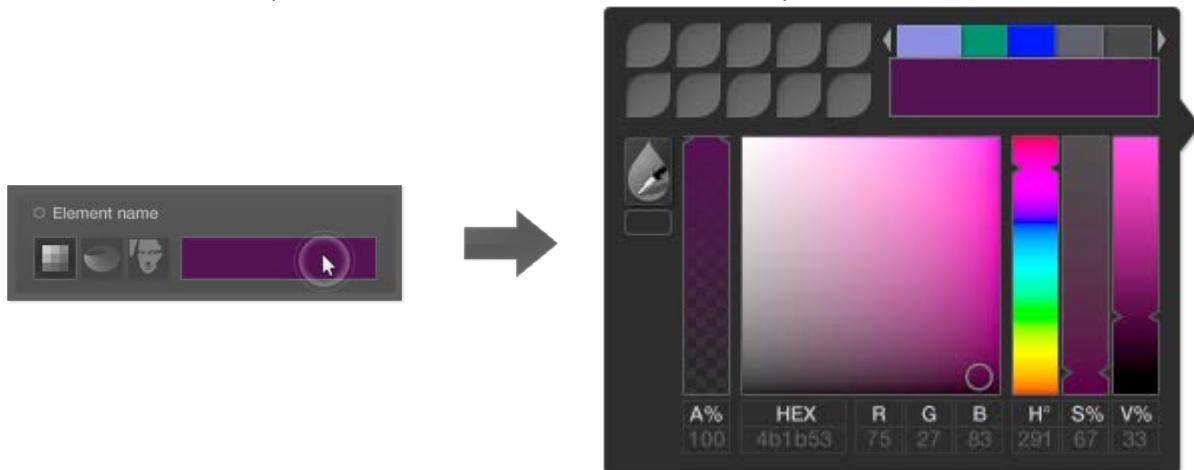


Classico colore a tinta unita

Tavolozza dei colori (Color palette)

Aprire la tavolozza

Quando fai clic sull'anteprima del colore, la tavolozza dei colori si apre:



Chiudere la tavolozza

Una volta che hai finito di impostare il colore, fai semplicemente clic in un punto qualsiasi all'esterno della finestra dei parametri del colore per chiudere la finestra e disegnare con quel colore.

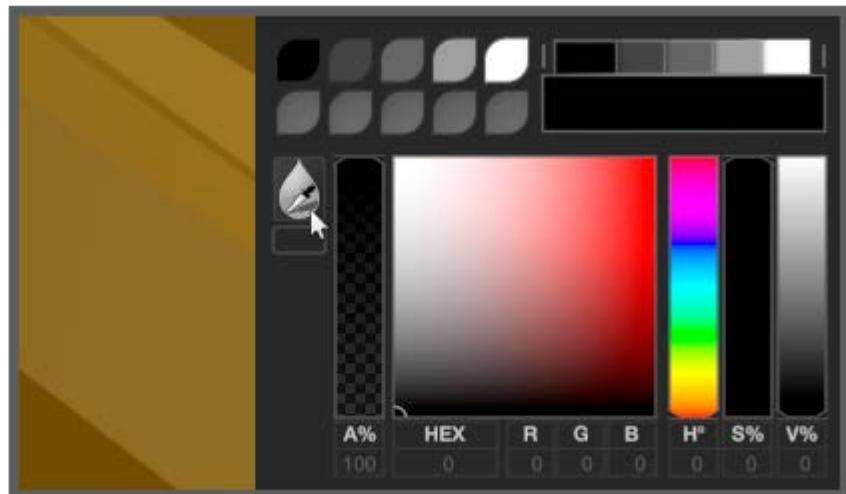
Cattura Colore (Eyedropper)

Utilizzo del Cattura Colore (contagocce)

Clicca sull'icona a forma di contagocce.

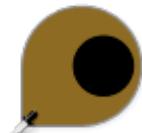
Il suo pulsante brilla e lo strumento è abilitato.

Se fai nuovamente clic, lo strumento viene disabilitato.

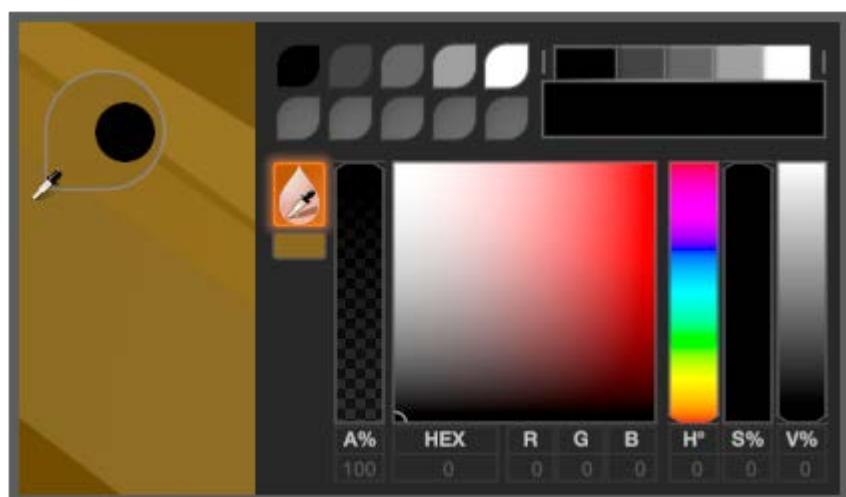


Trova il tuo colore

Quando il pulsante è evidenziato, se vai nell'area di disegno il cursore si trasforma in un contagocce.



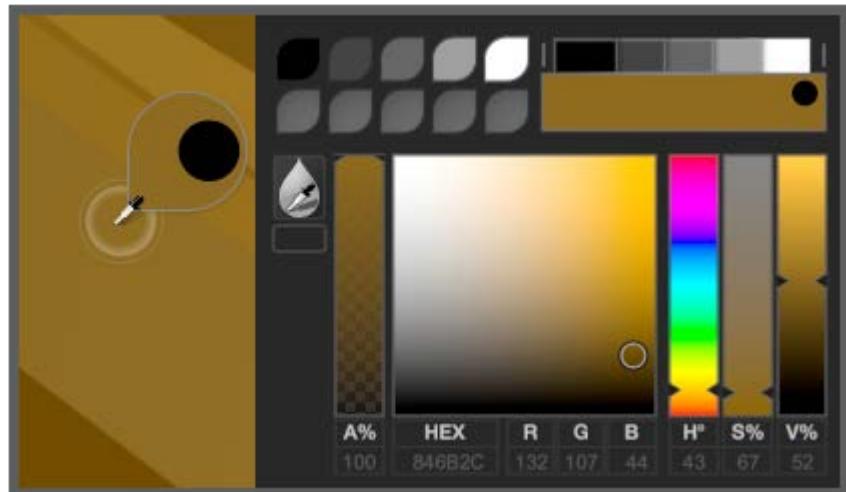
La forma a goccia è l'anteprima del nuovo colore. Il cerchio interno è il colore corrente.



Selezione il colore

Clicca con il contagocce sul tuo disegno, il colore è selezionato. Se hai scelto il colore sbagliato, fai nuovamente clic sull'icona del contagocce per ricominciare.

Disegnando nella tela, il colore della paletta si chiude e il nuovo colore è attivo.



Il contagocce prende solo i colori del tuo disegno e non i colori basati sull'immagine.

Alpha (Imposta la trasparenza del colore)

Il cursore a sinistra della palette imposta il valore alfa:



Il colore è opaco
Il suo valore alfa è al 100%



Il colore è trasparente
Il suo valore alfa è al 27%



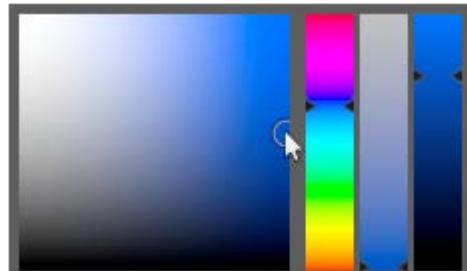
Il colore è invisibile
Il suo valore alfa è 0%

Impostazione visiva dei colori

Possiamo impostare un colore in maniera visiva (istintiva) in diversi modi:

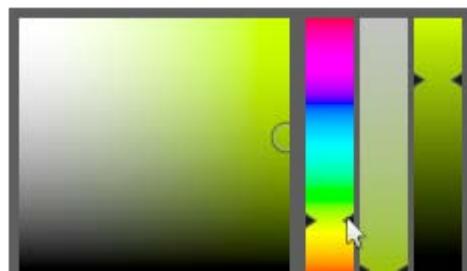
Nella tavolozza dei colori

Scegli direttamente nella tavolozza dei colori il suo valore e la sua saturazione.



Tinta

Cambia la tinta del colore



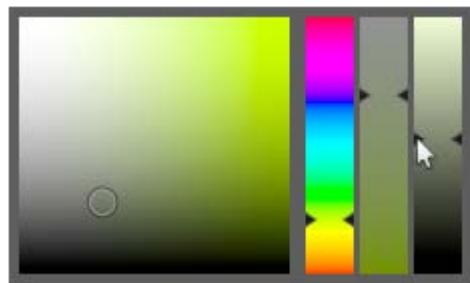
Saturazione

Rende il colore più grigio o più presente.



Valore

Rende il colore più chiaro o più scuro



Impostazione numerica dei colori (precisa)

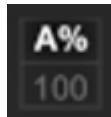
È possibile cambiare il tuo colore con un valore numerico.

Per modificare i valori numerici clicca sul valore desiderato o usa [tab] se ne hai già selezionato uno.

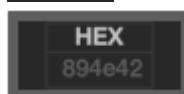
Inserire il valore desiderato e confermare con [invio].



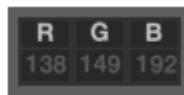
A% (Alpha)



HEX (Hexadecimal)



RGB (Red, Green, Blue)



HSV (Hue, Saturation, Value)



Colore sfumato (gradiente)



Il pennello prende i colori e le opacità definiti nel gradiente.

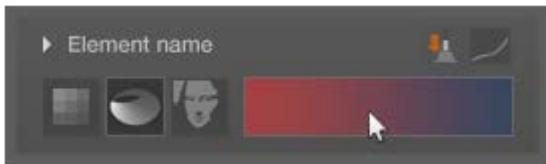
Questo tipo di colore è utile per:

- dipingere piume, capelli, tessuti, erba, pelliccia ...
- creare sfondi per cielo, paesaggi, ...
- portare una varietà di colori nel tuo tratto

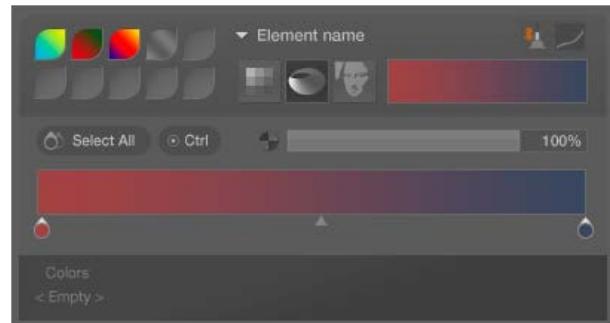
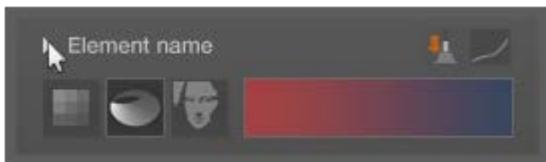
Editor dei gradienti

[Apertura dell'editor dei gradienti](#)

Clicca sull'anteprima del gradiente:



Oppure clicca sulla freccia prima del nome:



Quando torni al disegno, la versione compatta mostra solo le impostazioni essenziali.



Chiusura dell'editor dei gradienti

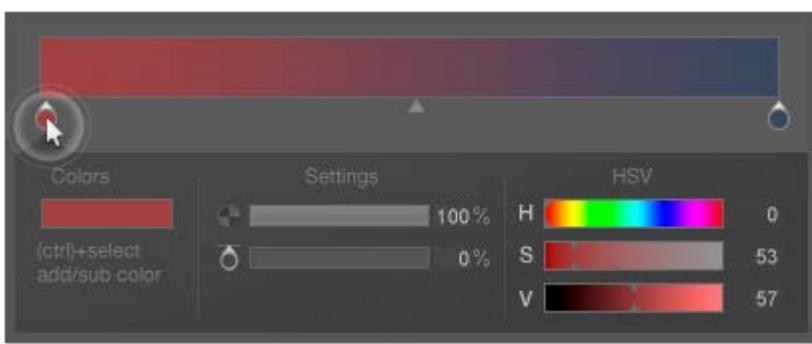
Dopo aver impostato la sfumatura, fai semplicemente clic in un punto qualsiasi all'esterno della finestra dei parametri di colore per chiudere la finestra e utilizzare queste impostazioni.

Creazione e modifica dei gradienti

Campioni

I gradienti (le sfumature) sono definite da “campioni” di colore.

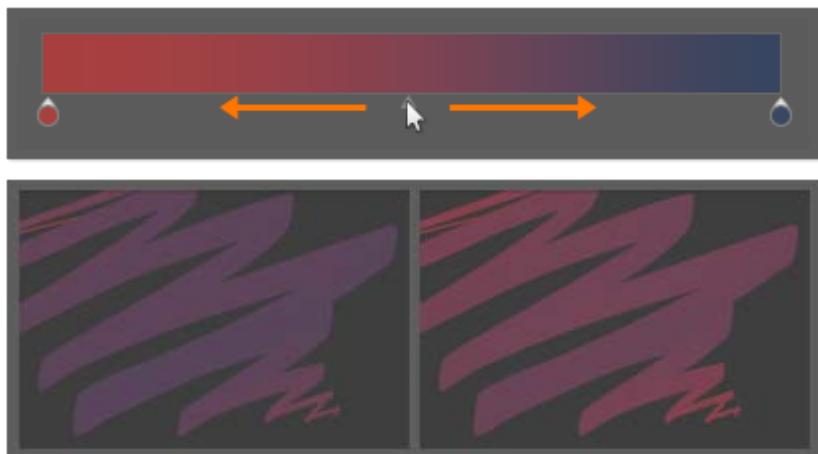
Il colore è interpolato automaticamente tra due campioni.



Aprire l'editor di selezione

Cliccando su un campione vengono visualizzati i suoi parametri.

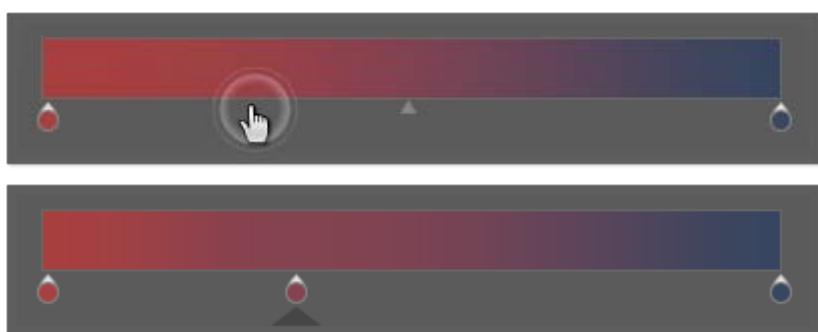
- Colori, per vedere il colore selezionato
- Impostazioni, per impostare la sua opacità e la sua posizione sul gradiente
- HSV, per regolare rapidamente il colore selezionato



Esempio sulla finestra di anteprima del pennello

Cambia il comportamento (linearità) del gradiente

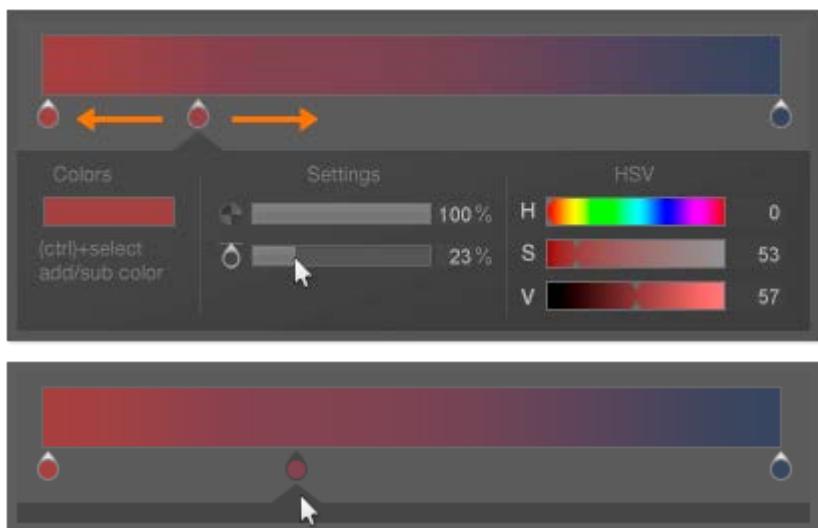
Quando metti il cursore tra due campioni appare un triangolo. Se lo si fa scorrere più o meno, il comportamento del gradiente cambia.



Aggiungere un campione

Posizionandosi vicino all'editor del gradiente viene visualizzato il cursore di puntamento (mano). In questa posizione è possibile aggiungere un campione di colori facendo clic.

NOTA: il campione aggiunto prende il colore intermedio visibile nel gradiente



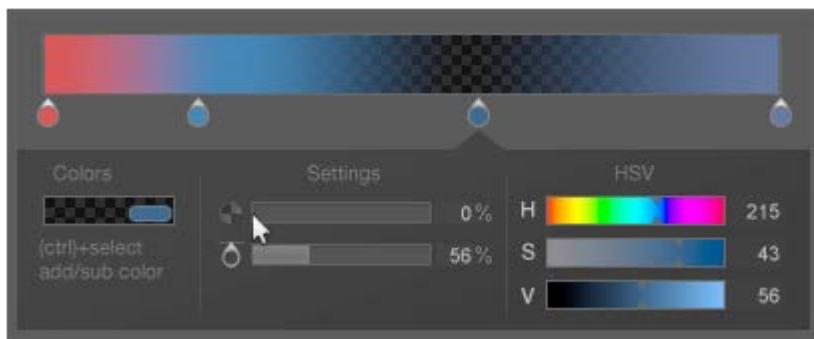
Spostare un campione

Prendi il campione e spostalo lungo il gradiente. Se si desidera un valore esatto, è anche possibile fare clic su di esso e modificarne il valore in Impostazioni (Settings).

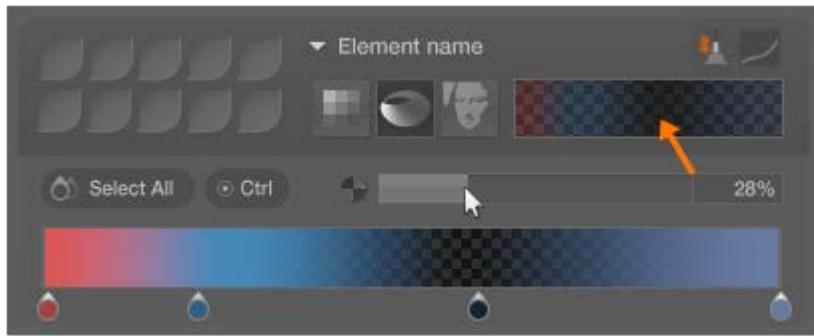
Eliminare un campione

Se clicchi e trascini un campione al di fuori dell'area, il suo aspetto cambia: se lo rilaschi, viene eliminato (drag & drop).

Il valore Alfa nei campioni e nei gradienti



Il valore Alfa di un campione
Nelle impostazioni del campione selezionato, se il suo alfa è diminuito, il gradiente ha una parte trasparente.



Il valore Alfa di un gradiente
Se diminuisci l'alfa globale del gradiente, allora il gradiente finale diventa più trasparente. È un modo rapido per cambiare la il valore di alfa (trasparenza) a tutto il gradiente.

Modificare contemporaneamente un gruppo di campioni

Selezione multipla dei campioni



Seleziona tutti i campioni
Per selezionare tutti i campioni contemporaneamente, fare clic su Selezione tutto

Tutti i campioni vengono modificati contemporaneamente quando si modificano i loro parametri.



Seleziona un gruppo di campioni
Per selezionare o deselectare ogni campione:

- Tenere premuto il tasto CTRL
- Attivare il pulsante Ctrl nell'interfaccia

Modifica di tutti i campioni selezionati

Lista dei colori dei campioni

Tutti i colori dei campioni selezionati appaiono in successione. Ogni anteprima di colore offre l'accesso alla cronologia dei colori.



Impostazioni

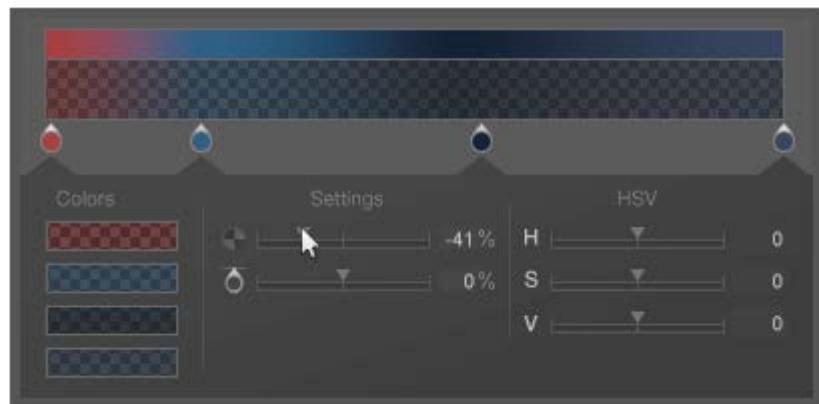
I campioni selezionati sono modificati proporzionalmente.



Alfa

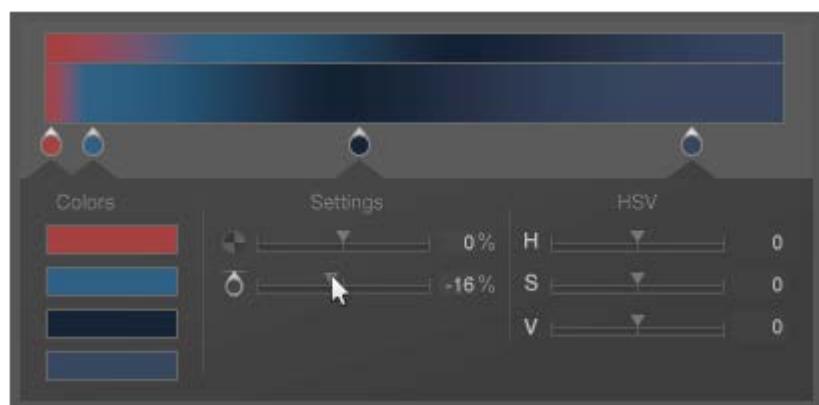
Quando si utilizza l'alfa sui campioni selezionati, questi vengono modificati in modo proporzionale:

- Se si diminuisce il valore alfa, diventano tutti più trasparenti.
- Se lo aumenti, diventano tutti più opachi.



Posizione

- Quando il valore della tua posizione è negativo, tutti i raccoglitori selezionati si trovano a sinistra del gradiente,
- Se è un valore positivo, vanno a destra del gradiente.

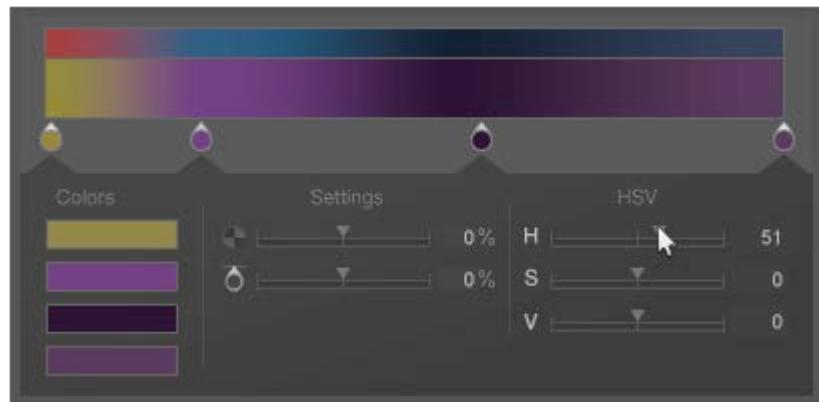


HSV



Tinta (Hue)

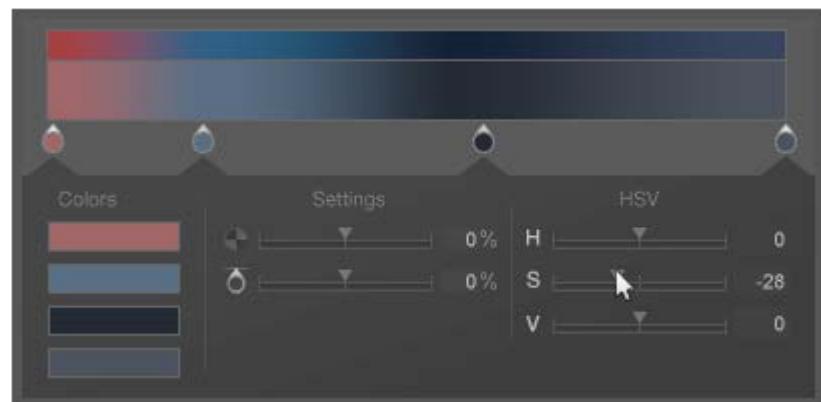
Permette di cambiare i colori dei campioni selezionati in base alla loro tinta originale.



Valore (Value)

Quando il valore di Saturazione è:

- Negativo il colore è più grigio.
- Positivo il colore è più presente.



Valore (Value)

Quando il valore V è:

- Negativo il colore è più scuro
- Positivo il colore è più leggero



Colore da immagine



Seleziona automaticamente i colori da un'immagine invece di sceglierli manualmente.

Questo tipo di colore è utile per:

- creare piacevoli disegni anche senza grandi capacità artistiche
- costruire e disegnare in base a un'immagine di riferimento.
- creare facilmente un insieme di colori realistici



Attualmente è possibile utilizzare solo immagini jpg e png.

Caricamento di un'immagine

Se non ci sono immagini caricate, fare clic sull'area di anteprima del colore.



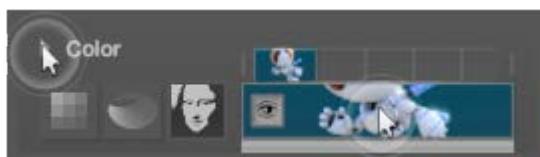
Viene visualizzata la finestra di dialogo Apri finestra del file. Scegli una foto sul tuo computer e fai clic su Apri.



Sostituzione dell'immagine esistente

Se è già presente un'immagine caricata, è necessario aprire l'editor dei colori basato su immagini.

Fare nuovamente clic sull'area di anteprima del colore o fare clic sulla freccia per aprire il relativo editor.



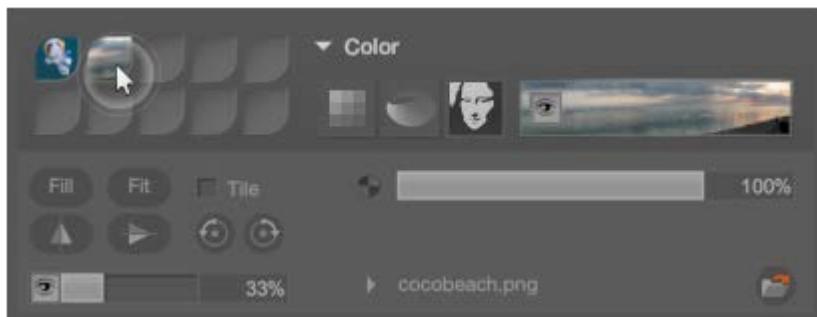
Viene visualizzata la finestra di dialogo Apri file finestra e puoi cambiare la tua immagine con un'altra.



Se invece l'editor è aperto, fai clic su



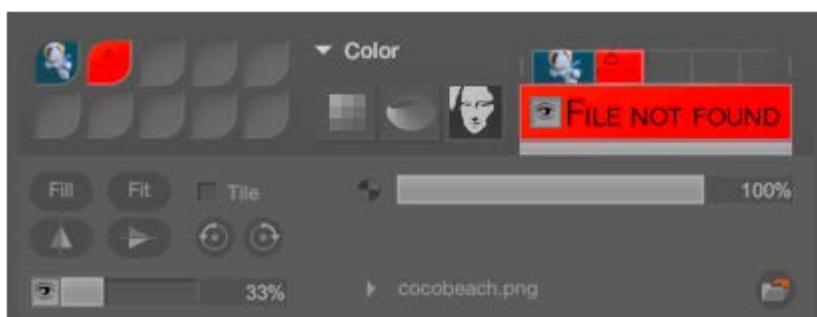
Caricare un'immagine dai colori preferiti



Colori preferiti

Posiziona il mouse sopra l'anteprima del colore e si aprirà il pannello.

Poi clicca sui colori preferiti per selezionare la tua immagine.



L'immagine non è stata trovata

A volte l'immagine preferita è stata spostata o eliminata dalla cartella iniziale.

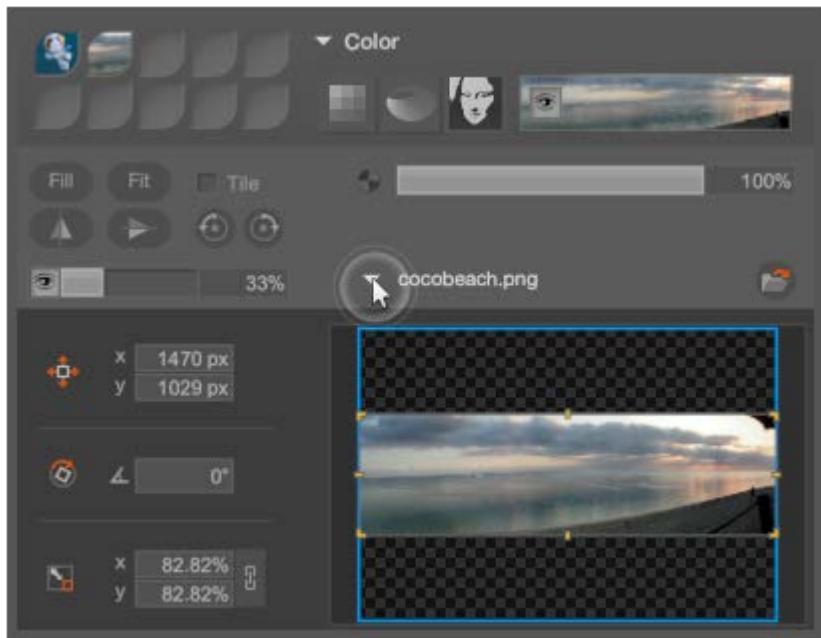
"File not found" (File non trovato) è mostrato in anteprima.

Fai clic sull'icona del file aperto per controllare manualmente il computer.

Fare clic sulla freccia accanto al nome dell'immagine per aprire il pannello e trasformare l'immagine corrente.

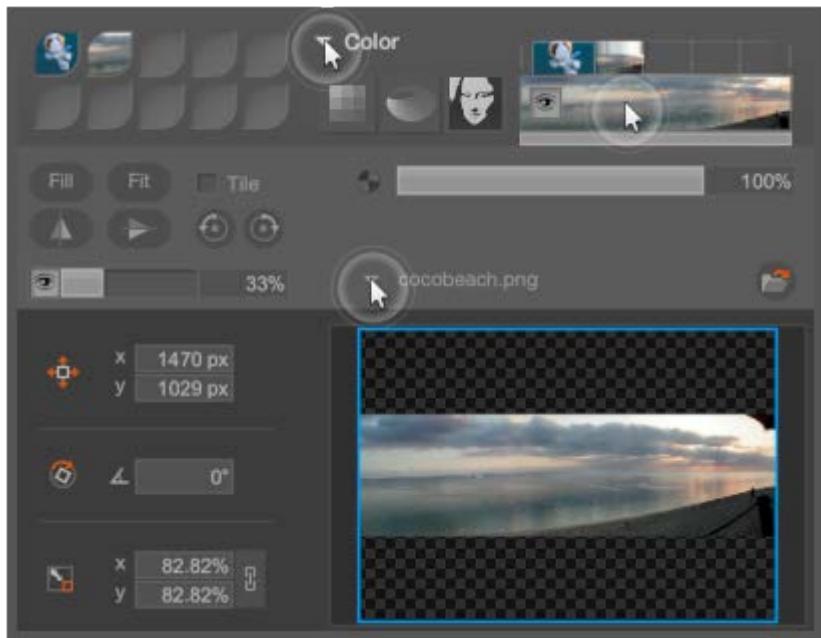
Modifica l'immagine caricata

- La parte sinistra mostra informazioni sulla posizione, la rotazione e la scala (uniforme o non uniforme). Puoi anche inserire valori usando la tastiera.



▪ Sul lato destro troviamo la tela che è rappresentata da una cornice blu e al suo interno l'immagine. Puoi trasformare direttamente la tua immagine tramite gli ancoraggi gialli.

Image Editor compatto



Basta fare nuovamente clic sugli elementi utilizzati per compattare l'interfaccia.

Se si fa nuovamente clic, l'interfaccia si aprirà completamente.

Posizionamento e visualizzazione dell'immagine

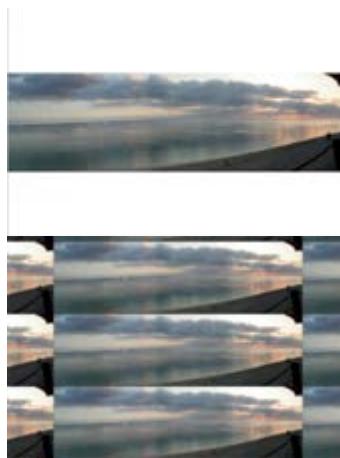
Fill (Riempe)

L'immagine viene scalata per entrare interamente nella tela sia in altezza che in lunghezza.



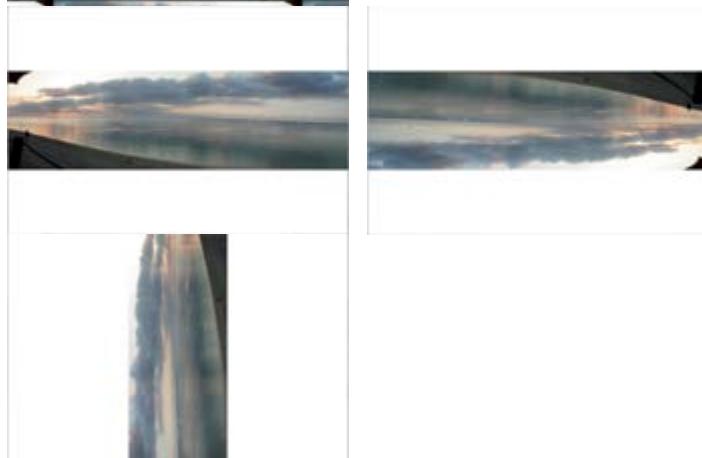
Fit (adatta)

L'immagine viene regolata per entrare interamente nella tela o in altezza o in lunghezza.



Tile (adiacente)

La stessa immagine si moltiplica per coprire la tela.



Flip (capovolge)



L'immagine viene capovolta orizzontalmente o verticalmente.



Rotate 90° (ruota)



L'immagine viene ruotata di 90° a destra o a sinistra.



Visualizzazione immagine



Questa visualizzazione utile per vedere dove sono i dettagli dell'immagine. Non è usato dal pennello o dal contagocce.



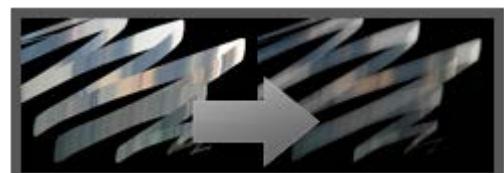
Modifica la visualizzazione dell'immagine (più o meno trasparente)

Clicca su questa icona per mostrare/nascondere l'immagine.

Alfa del colore



Questo cursore imposta la trasparenza del colore (quella del pennello, non dell'immagine).



Spostamenti precisi

Usa i campi numerici per inserire valori esatti con la tastiera.



Move (sposta)



Rotate (ruota)



Uniforme

Non uniforme

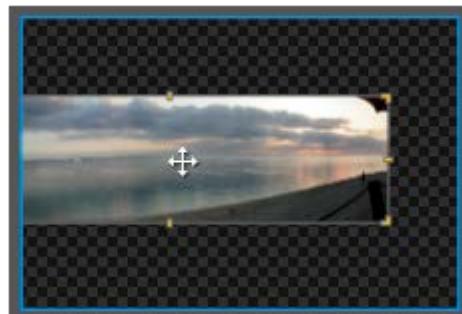
Scale (scala)

Spostamenti a mano libera

Modifica l'immagine direttamente dall'editor.

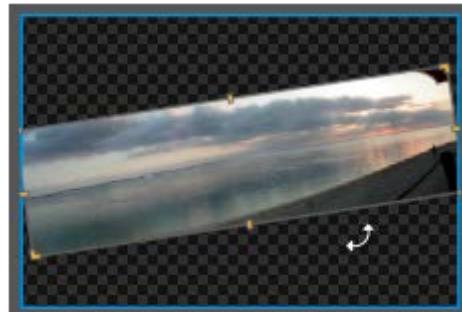
Move

Prendi l'immagine e muovila.



Rotate

Posizionandoti esternamente all'immagine la puoi ruotare.



Uniforme Scale

Trascinando un angolo ottieni una scalatura uniforme.



Non-uniform Scale

Trascinando un lato ottieni una scalatura non uniforme.



Cursori (Slider)

Si tratta di dispositivi di scorrimento che permettono di modificare il valore di un parametro.

Cursori semplici

Il semplice cursore modifica il valore massimo che il parametro può raggiungere durante la pennellata.



È possibile modificare manualmente il valore massimo del cursore facendo clic sul valore e digitando un numero o trascinando orizzontalmente:



Nota: l'opacità non può essere impostata a più del 100%

Cursori Estesi

Fai clic sulla freccia accanto al nome del parametro per aprire una versione estesa del cursore:



Nella versione estesa è possibile modificare i valori di minimo e di massimo:



Il valore minimo è il valore più basso che il parametro può avere durante il tratto del pennello:



In questo esempio, la forma non sarà mai più piccola di 7 pixel quando la pressione è bassa e mai più grande di 23 pixel quando si preme più forte sullo stilo:



Puoi modificare entrambi i valori minimo e massimo allo stesso tempo spostando il piccolo blocco sopra il cursore:



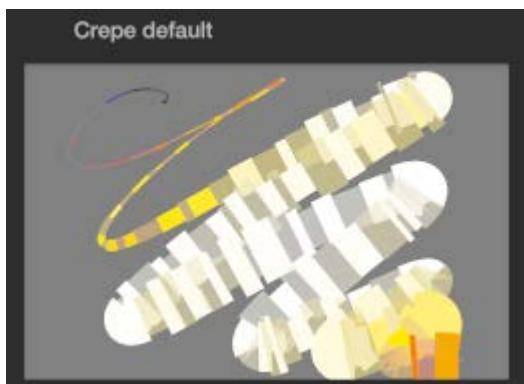
Questo modificherà i due valori mantenendo la loro differenza relativa.

Impostando lo stesso valore su minimo e massimo si disabilita il controller e il parametro avrà sempre lo stesso valore (es. la dimensione non cambierà rispetto alla pressione dello stilo):



Segmenti (Segments)

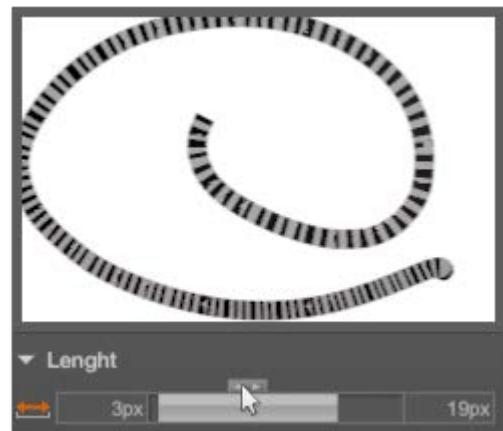
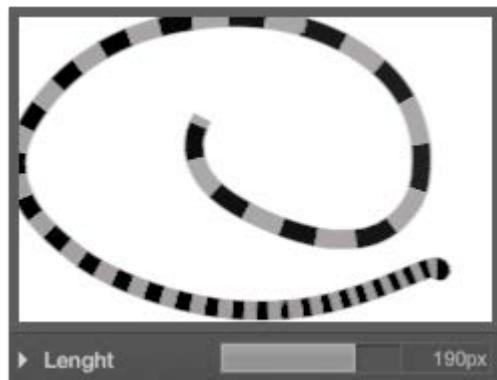
Questo pannello viene visualizzato solo dal pennello **Crepe default** (e dai pennelli che potete creare partendo da questo):



Parametri caratteristici:

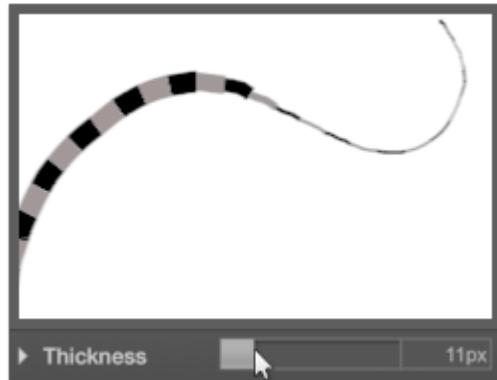
Length
(Lunghezza)

Cambia la
lunghezza del
segmento.



Thickness
(Spessore)

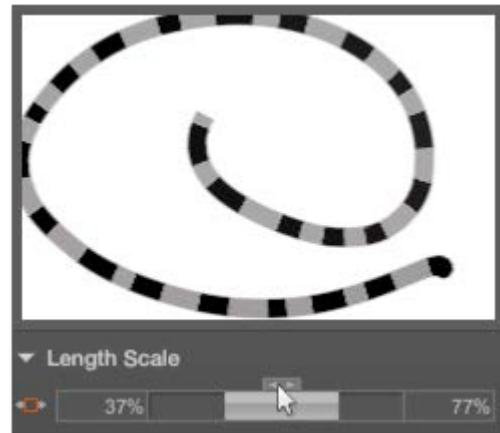
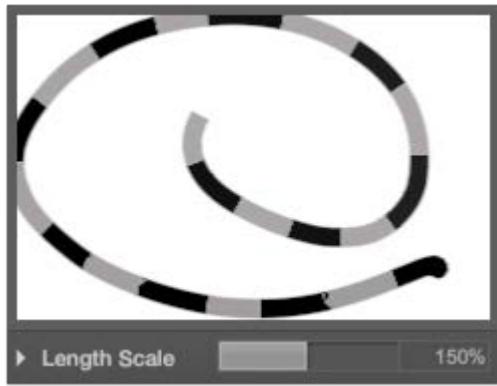
Cambia la
larghezza del
segmento.



Length Scale
(Scala
Lunghezza)



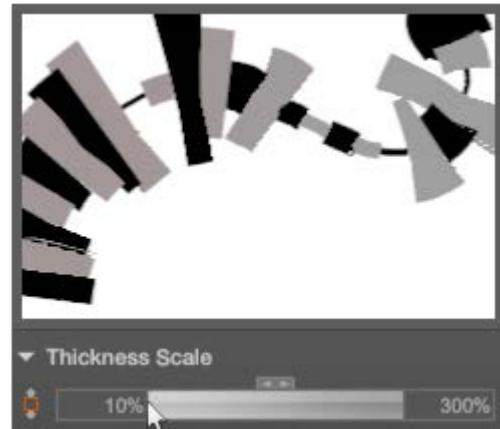
Aggiunge un fattore casuale alla lunghezza del segmento.



Thickness Scale (Scala
Larghezza)



Aggiunge un fattore casuale alla larghezza del segmento.



Forma del pennello (Shapes)

Nel pannello **Shapes** è possibile modificare l'aspetto e il comportamento del pennello.

Molti pennelli condividono gli stessi parametri standard come **Size (Dimensione)**, **Opacity (Opacità)**, **Orientation (Orientamento)** e **Scale (Scalatura)**.

Size (Dimensione)



Cambia lo spessore del tratto.



Opacity (Opacità)



Cambia l'opacità del tratto.



Rotation (Rotazione)



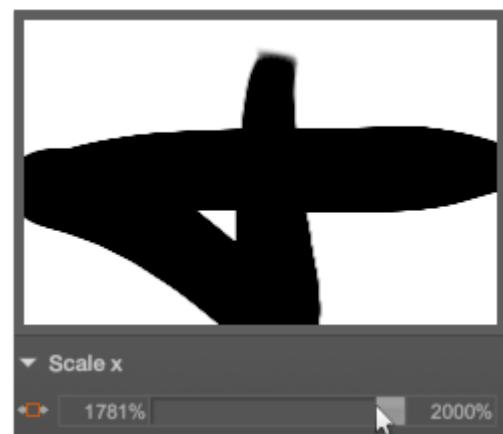
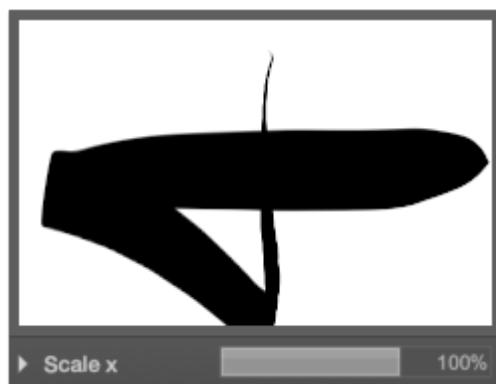
Definisce quanto ruota il pennello durante la pennellata.



Scale X
(Scala X)



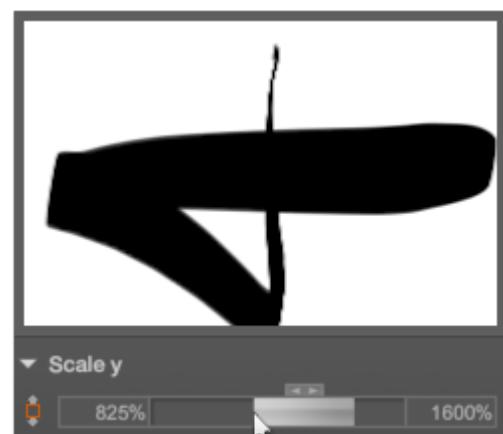
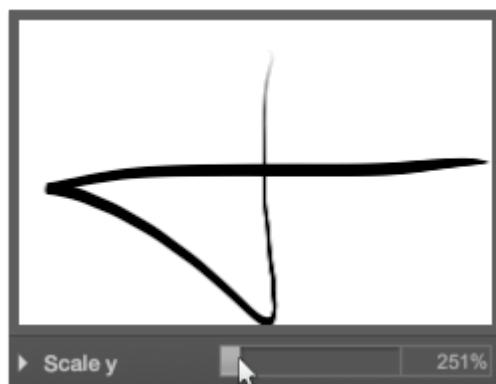
Cambia la scala
orizzontale
della forma.



Scale Y
(Scala Y)



Cambia la scala
verticale della
forma.

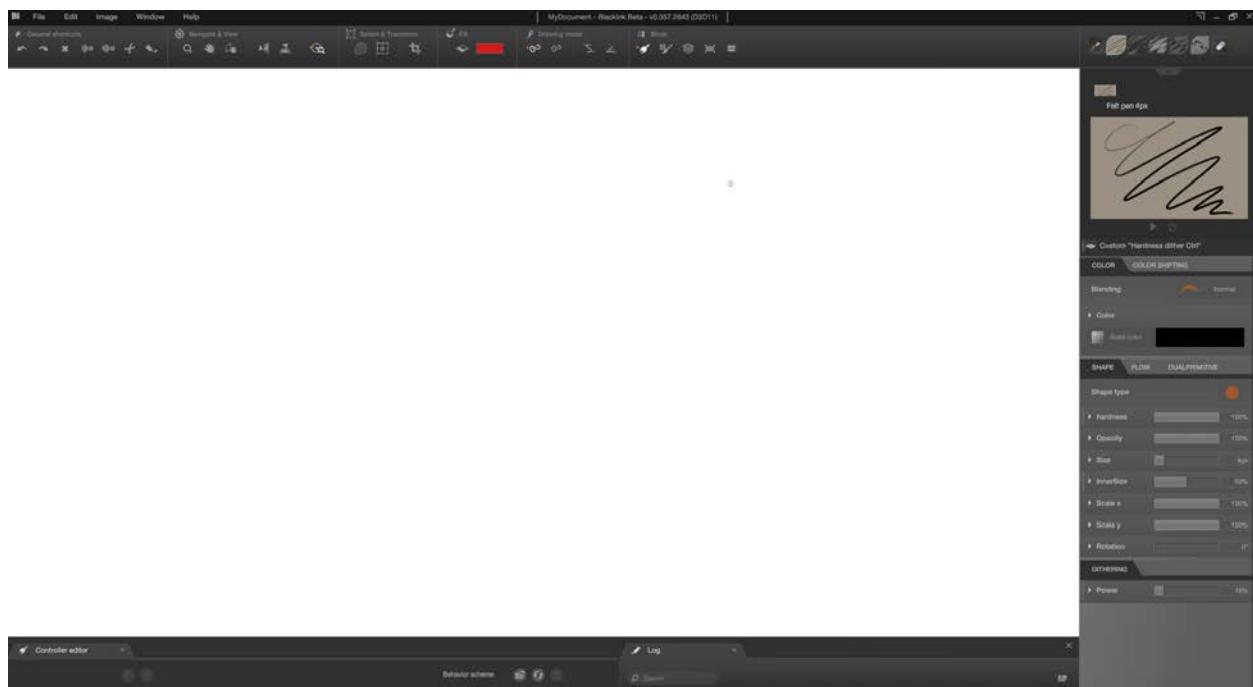


Tutorial: Primi passi

Ciao! Se sei un nuovo utente di BlackInk, sei nel posto giusto!

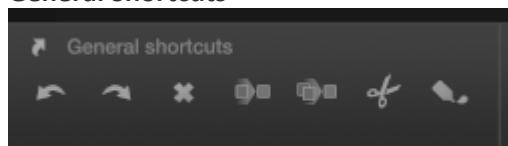
In questo tutorial ci accingiamo ad avere il nostro primo sguardo all'interfaccia utente e agli strumenti di base di BlackInk che ci permetterà di iniziare a disegnare fin da subito!

Al primo avvio BlackInk presenta la seguente finestra:



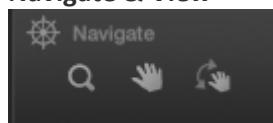
Cominciamo il tour in alto a sinistra. Appena sotto il menu standard è possibile vedere una serie di pannelli:

General Shortcuts



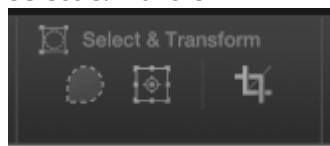
Comandi generali da tastiera (**Undo**, **Redo**, **Clear**, **Copy**, **Copy merged**, **Taglia**, **Incolla**)

Navigate & View



Comandi di Navigazione e Visualizzazione (**Zoom**, **Move**, **Rotate**, **Flip H/V**, **Constant brush size**)

Select & Transform

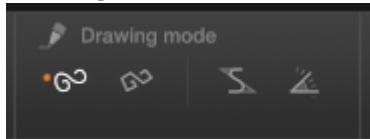


Fill



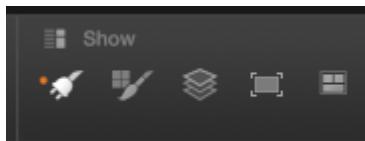
Comandi di Riempimento livello (**Fill, Color**)

Drawing mode



Comandi per i parametri del disegno (**Freehand, Straightline, Smoothness, Angle snap**)

Show

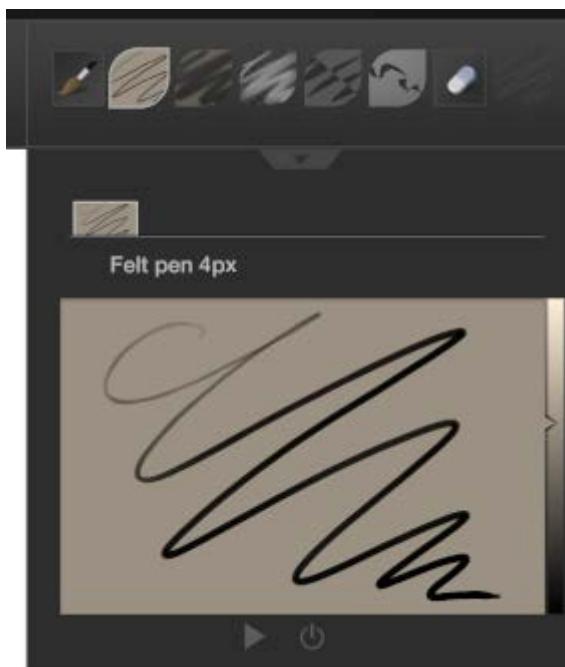


Comandi per la visualizzazione delle finestre (**Controller editor, Brush Manager, Layer Manager, Secondary View, Docked panels**)

Alcuni di questi comandi hanno delle opzioni che si aprono quando si posiziona il cursore su di loro e mostrano le loro scorciatoie da tastiera (“**Q**” -> zoom, “**SPACE**”-> Sposta, “**R**”->Ruota, ecc.) Questi ultimi lavorano tenendo premuto il tasto e spostando il mouse. Quando hai finito, basta rilasciare il tasto. La maggior parte dei comandi ha un significato intuitivo e basta provarli per rendersi conto del loro funzionamento.

Pannello dei Pennelli

In alto a destra si può vedere un set di pennelli predefiniti:

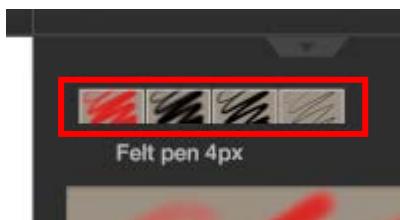


Essi sono: **Felt pen 4px, Airbrush 12px, Smudge colored, Free fill, Angle line fill e Eraser.**

Provali, ognuno ha un comportamento molto diverso.

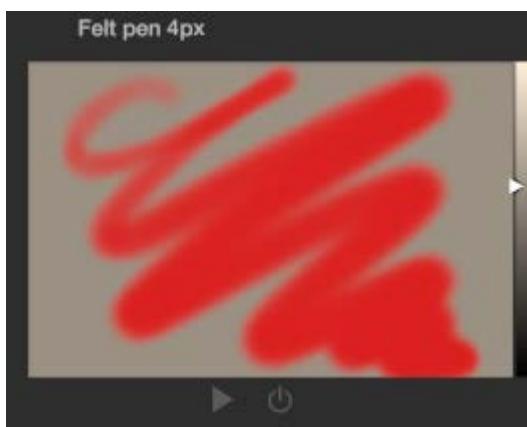
Lo strumento **Gomma (Eraser)** si può attivare con il tasto "E" e poi tornare al pennello selezionato in precedenza con il tasto "B" (oppure "di nuovo "E").

Se si modificano una o più impostazioni del pennello (per esempio la dimensione) viene creato un nuovo pennello accanto al precedente con le nuove impostazioni.



Questa è la **"Storia del Pennello"** ed è molto utile. Non vi resta che selezionare una delle impostazioni precedenti per richiamare istantaneamente il relativo pennello.

Sotto la storia dei pennelli troviamo la finestra di anteprima del pennello attivo:



In questa finestra viene visualizzata un'anteprima del tratto del pennello di default che riflette istantaneamente ogni modifica apportata nelle impostazioni.

Provate a modificare le dimensioni o l'opacità, si vedrà l'aggiornamento dell'anteprima in tempo reale. Potete anche disegnare la vostra pennellata sulla finestra di anteprima.

Il pulsante ripete la pennellata.

Il pulsante ripristina la pennellata di default.

Lo freccia bianca scorrevole (slider) che si trova a destra della finestra di anteprima permette di modificare il colore di sfondo.

Brush Pixel shader editor

Appena sotto la finestra di anteprima troviamo un pannello che ci permette di caricare o definire (programmare) lo **Shader del Pennello**.



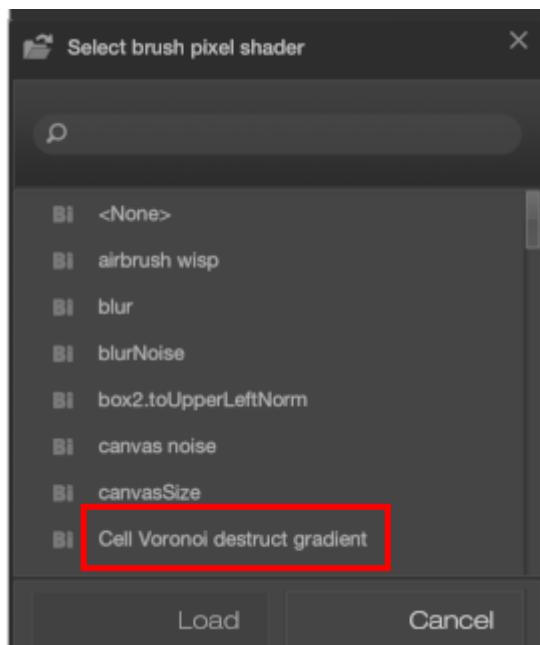
Il pulsante  ci permette di caricare un shader esistente.

Il pulsante  apre l'editor dello shader con il codice dello shader attivo.

Il pulsante  blocca la modifica accidentale

Per programmare uno shader occorre una certa esperienza, però possiamo caricare uno shader esistente per vedere che cosa accade al pennello.

Selezioniamo il pennello **Felt pen 4px**. Vediamo che lo shader predefinito per questo pennello è “**Hardness dither Ctrl**”. Adesso premiamo il pulsante  e carichiamo lo shader “**Cell Voronoi destruct gradient**”:



Adesso se disegniamo con questo pennello otteniamo qualcosa del genere:

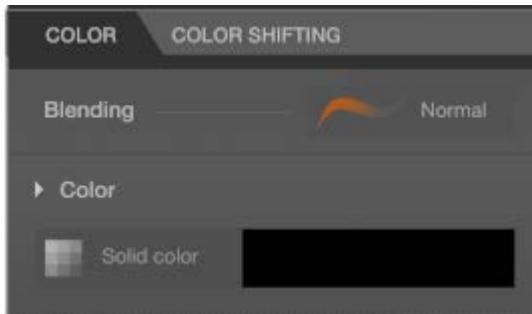


I pannelli dei Parametri

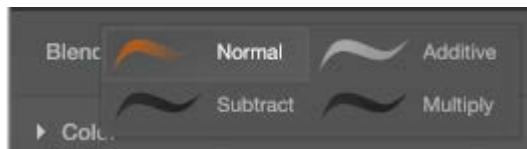
E poiabbiamo i parametri dei pennelli. Ci sono tre pannelli diversi che analizzeremo separatamente:

1. COLOR – COLOR SHIFTING
2. SHAPE – FLOW – DUAL PRIMITIVE
3. DITHERING

COLOR – Colore



In alto abbiamo il parametro **Blending** che ci permette di selezionare uno dei seguenti tipi:

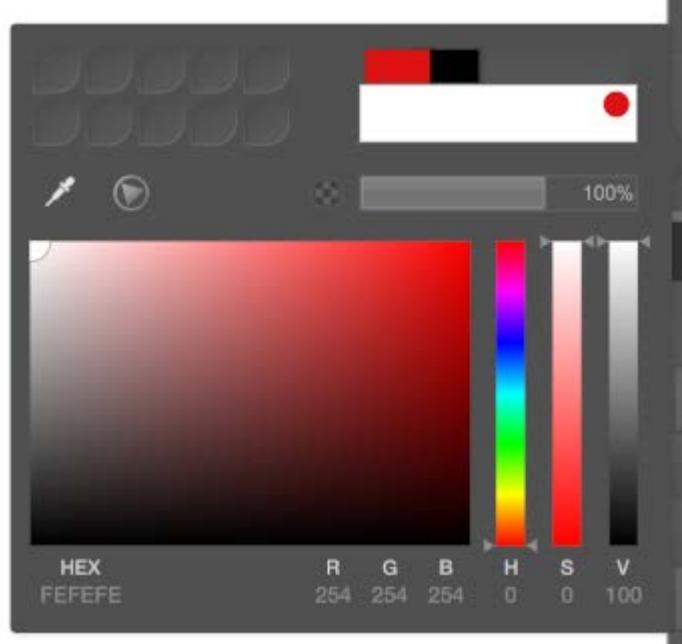


Questo parametro cambia il modo in cui il colore viene applicato quando si disegna (provate).

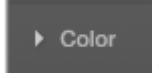
Per **selezionare un colore** è sufficiente cliccare il rettangolo grande colorato:

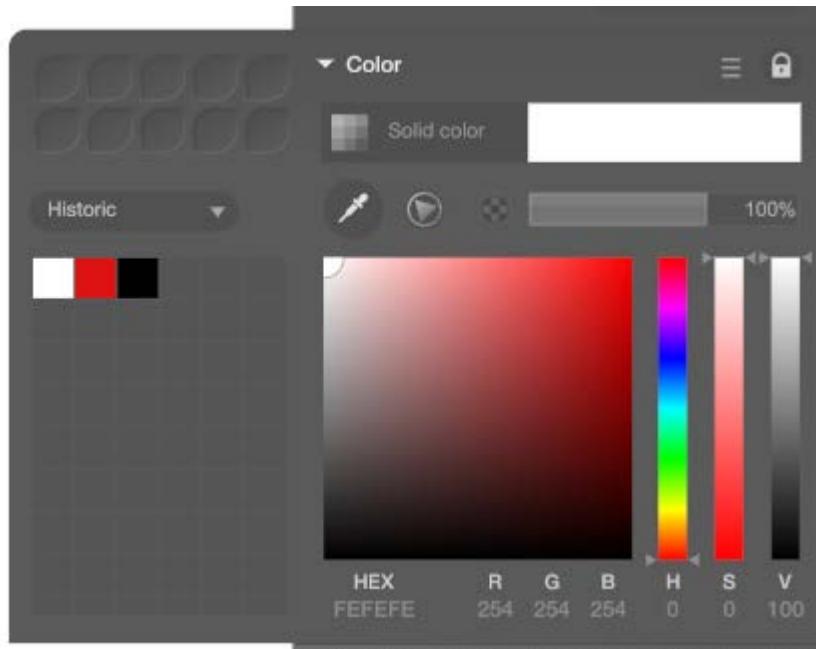


In questo caso appare la seguente finestra:



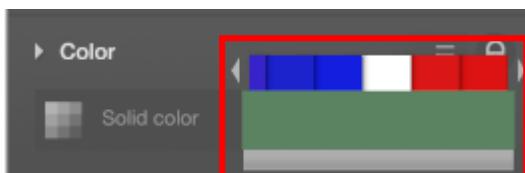
Con cui possiamo scegliere il colore preferito in diversi modi.

Il controllo completo del colore si ottiene premendo il pulsante  che apre la seguente finestra:



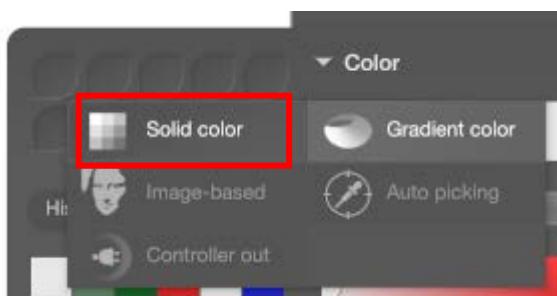
In cui troviamo la **Storia dei colori** (a sinistra).

Da notare che la storia dei colori è disponibile anche dall'interfaccia principale, semplicemente posizionando il mouse sopra il rettangolo grande che rappresenta il colore:



Modalità di colore

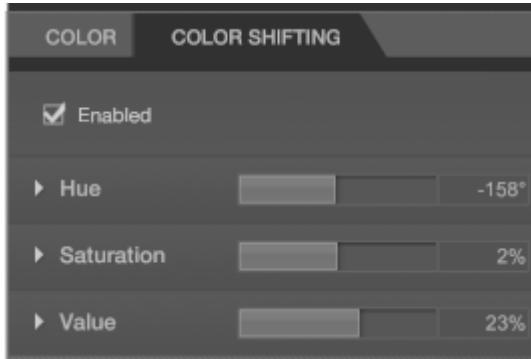
Ci sono cinque modalità colore: **Solid color**, **Gradient color**, **Image-based**, **Auto picking**, **Controller out**. Per accedere a questi parametri occorre premere il pulsante **Solid color**:



Queste modalità sono strumenti molto interessanti e potenti che saranno analizzati in tutorial successivi.

COLOR-SHIFTING

Questo pannello permette di modificare (shiftare) il colore attivo utilizzando **Hue**, **Saturation** e **Value**:



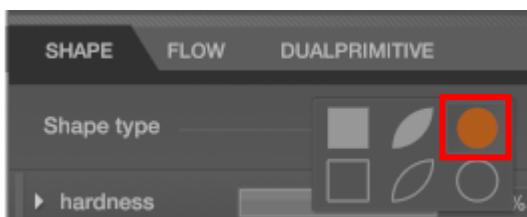
In pratica ci permette di avere due colori per il colore attivo. Oppure ci permette di scegliere un colore simile a quello attivo basandoci su variazioni di **Tinta (Hue)**, **Saturazione (Saturation)** e **Luminosità (Value)**.

SHAPE – FLOW – DUAL PRIMITIVE

SHAPE

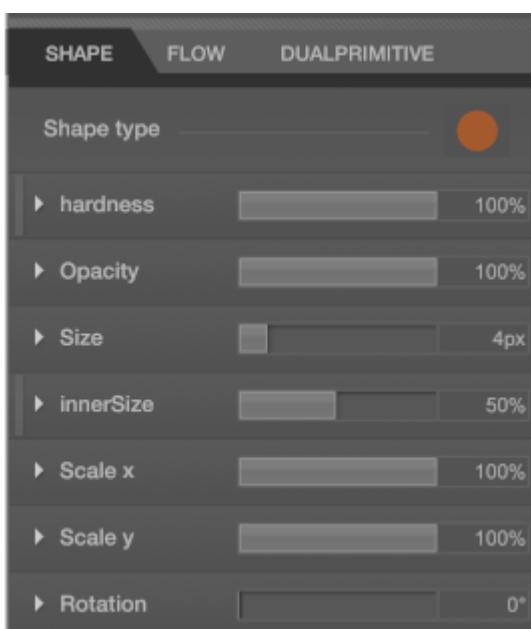
In questa sezione troviamo le impostazioni tipiche del pennello.

In alto troviamo il parametro **Tipo di forma (Shape type)** a cui si accede premendo il pulsante evidenziato:



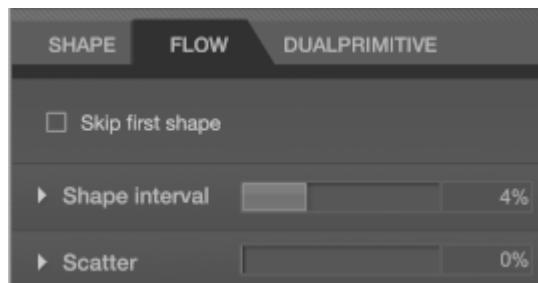
Questo parametro cambia la forma utilizzata per creare il tratto del pennello. Questo può cambiare completamente l'aspetto del nostro pennello.

Poi troviamo i parametri per la **durezza (Hardness)**, l'**opacità (Opacity)** la **dimensione (Size)**, la **dimensione interna (innerSize)**, la **scala X (Scale x)**, la **scala Y (Scale y)** e la **rotazione (Rotation)**:



Provare a cambiare alcuni valori e verificate come cambia il tratto del pennello (notare anche che le variazioni si riflettono istantaneamente sulla finestra di anteprima del pennello).

FLOW



In questo pannello troviamo due parametri: **Shape interval** e **Scatter**

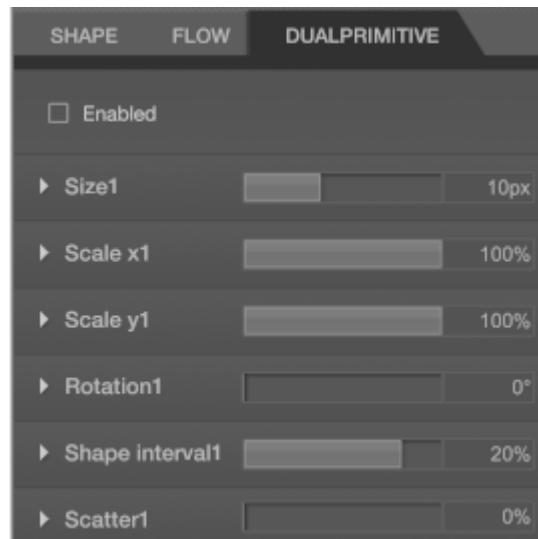
“**Shape interval**” imposta la distanza tra le forme che compongono la pennellata, in questo modo si possono disegnare forme separate invece che continue.

“**Scatter**” (**Sparpagliamento**) controlla come il colore debba seguire la pennellata.

Ecco alcuni esempi con il pennello **Fet pen 4px** con valori diversi per i due parametri:

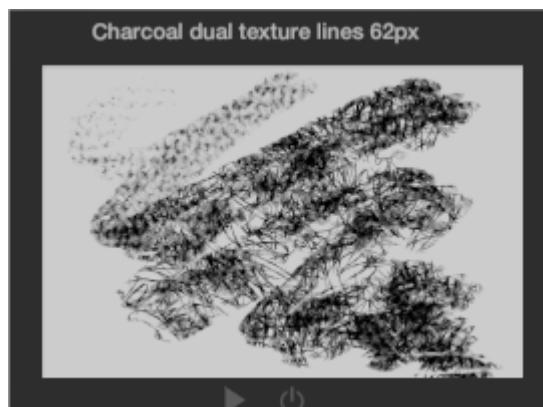


DUAL PRIMITIVE



Dual Primitive è una seconda serie di coordinate primitive generate per ogni primitiva visualizzata durante i passi necessari alla creazione della forma da parte del motore del pennello (brush engine). Generalmente il **Brush Shader Language** recupera il secondo set di parametri per calcolare un'altra posizione della texture.

Ad esempio, il pennello **Charcoal dual Texture lines 62px** usa questo parametro:



DITHERING

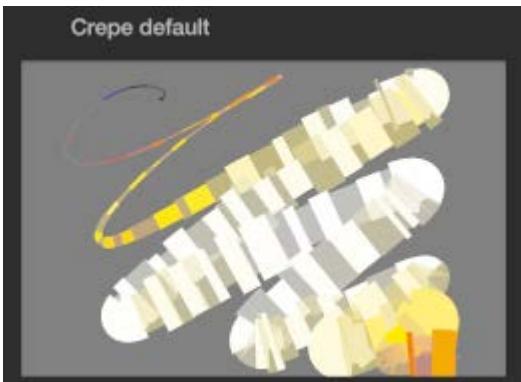


Il parametro **“Power”** del pannello **DITHERING** controlla il “tremolio” della pennellata. Come al solito per capire meglio di cosa si tratta occorre fare delle prove.

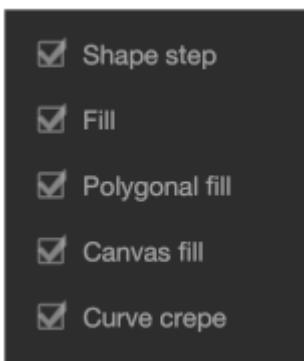
SEGMENT



Questo pannello è disponibile solo con il pennello **Crepe default**:



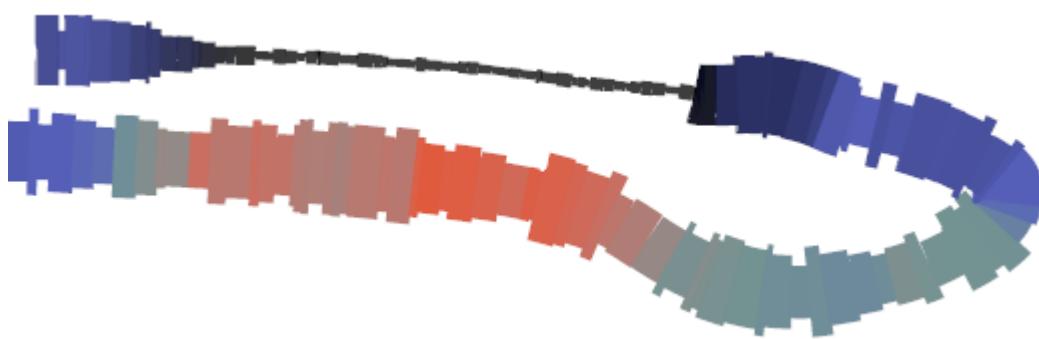
In effetti i pennelli di BlackInk hanno un diverso motore di rendering, puoi vederli nel gestore dei pennelli (M) dove puoi filtrare i tuoi pennelli:



I parametri del pennello del segmento sono solo per il pennello predefinito di Crepe.

Naturalmente è possibile creare più pennelli con il linguaggio Brush Shader a partire dalla Crepe.

Per capire i parametri del pennello Crepe default vediamo una pennellata:



I parametri si riferiscono ai segmenti (rettangoli) che compongono la pennellata:

Opacity: imposta l'opacità del segmento

Length: imposta la lunghezza del segmento

Length Scale: imposta la scala della lunghezza del segmento

Thickness: imposta lo spessore (larghezza) del segmento

Thickness Scale: imposta lo spessore (larghezza) del segmento

I CURSORI

Ora che abbiamo visto le parti principali del nostro schermo di lavoro, parliamo di un'ultima cosa importante: **i cursori**.

Essi sono utilizzati per modificare le dimensioni e l'opacità e tutti i tipi di parametri ed è molto importante

capire come funzionano.

L'unica cosa da sapere è che hanno due stati: **semplice ed esteso**.

Stato semplice



Stato esteso



Per estendere un cursore, fare clic sulla piccola freccia bianca accanto ad esso.

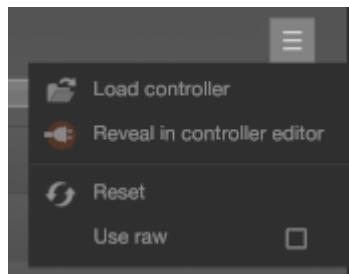
Allora qual è la differenza tra questi due stati?

Nella versione estesa di un cursore, è possibile impostare un valore minimo e un valore massimo. Questo perché spesso si desidera un valore di cambiare secondo certe regole. Ad esempio, quando la dimensione è impostata per seguire la pressione del tablet, il minimo sarebbe la dimensione quando non si preme e il massimo sarebbe la dimensione quando si preme al massimo.

Ora, quando si comprime il cursore alla versione semplice, le modifiche apportate non mancherà di tenere quella relativa gamma e applicarlo ai valori più alti o più piccoli. Quindi, in pratica, il dispositivo di scorrimento esteso viene utilizzato per impostare l'intervallo del vostro valore e la semplice è più veloce per regolare il risultato finale.

I cursori estesi hanno valori massimi predefiniti, ma se non è sufficiente per l'impostazione, è possibile trascinare il numero e cambiare il loro intervallo.

Premendo il tasto si apre una finestra che permette di gestire il parametro tramite i controller:



Questo argomento sarà trattato nelle prossime lezioni.

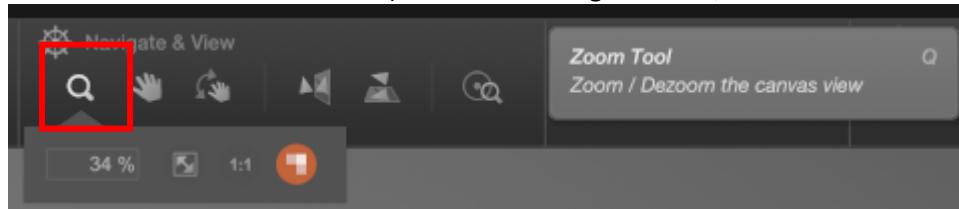
E questo è tutto per il nostro primo sguardo a BlackInk.

Provate i diversi pennelli predefiniti, modificate i parametri di base e divertitevi disegnando!

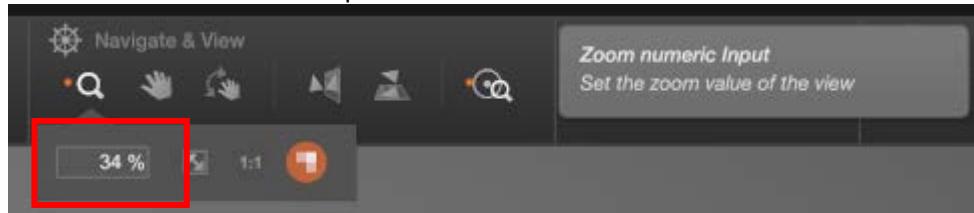
Tutorial: Navigazione e Visualizzazione (Navigate & View)

ZOOM

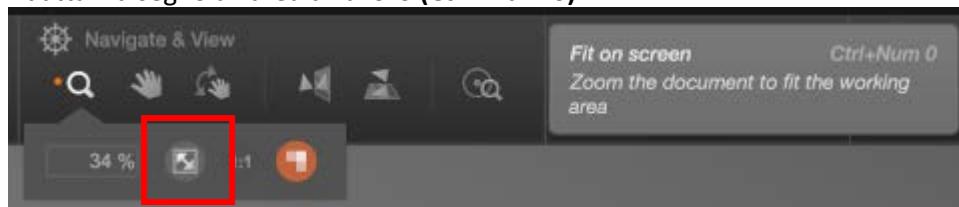
Premere **Q** e muovere il mouse (verso destra->Ingrandisce, verso sinistra->Riduce)



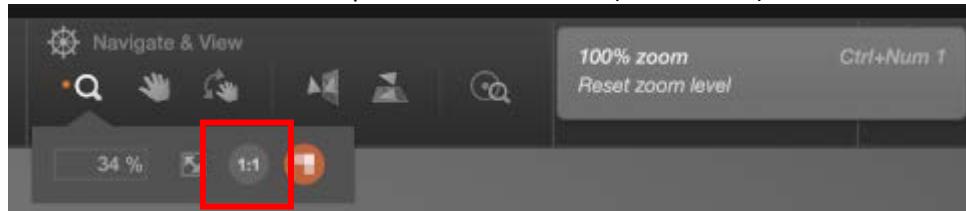
Inserire un valore numerico per lo zoom



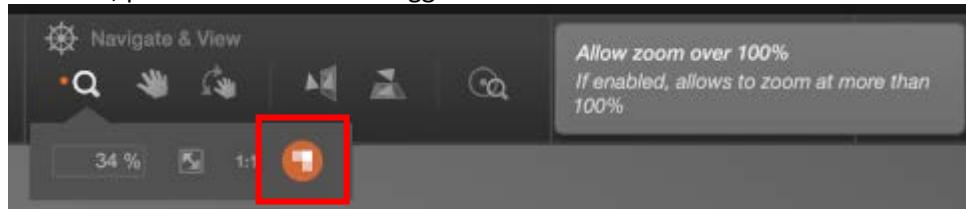
Adatta il disegno all'area di lavoro (**Ctrl+Num 0**)



Zoom al 100% centrato sulla posizione del cursore (**Ctrl+Num 1**)



Se attivo, permette uno zoom maggiore del 100%

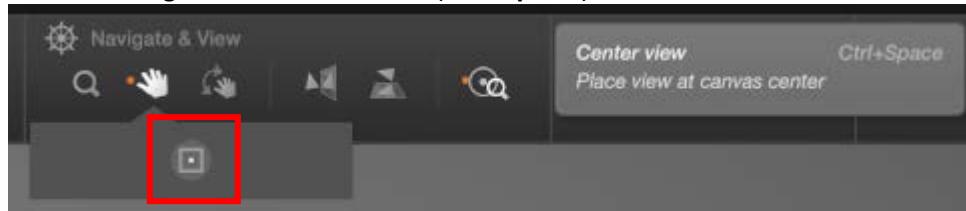


SPOSTAMENTO

Premere **SPAZIO** e muovere il mouse per spostare il disegno

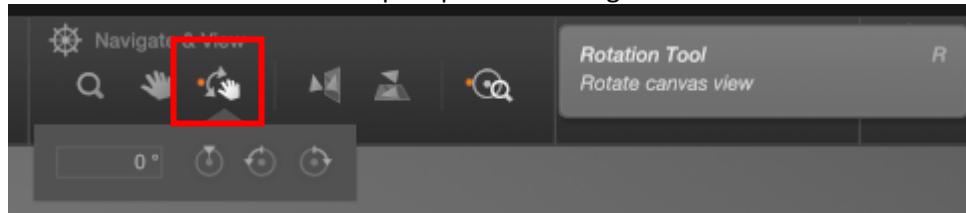


Centra il disegno nell'area di lavoro (**Ctrl+Spazio**)

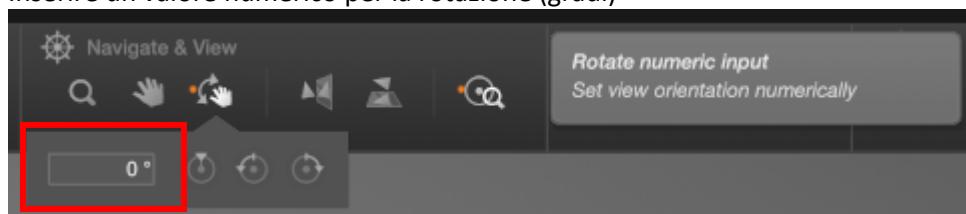


ROTAZIONE

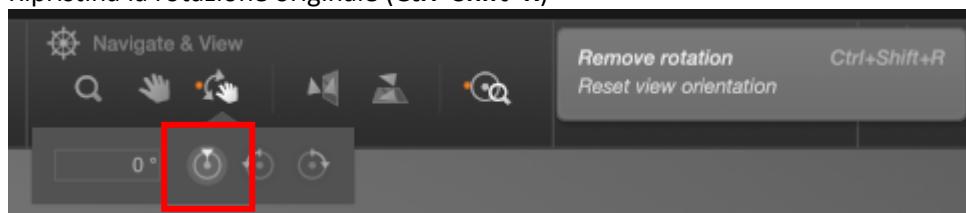
Premere R e muovere il mouse per spostare il disegno



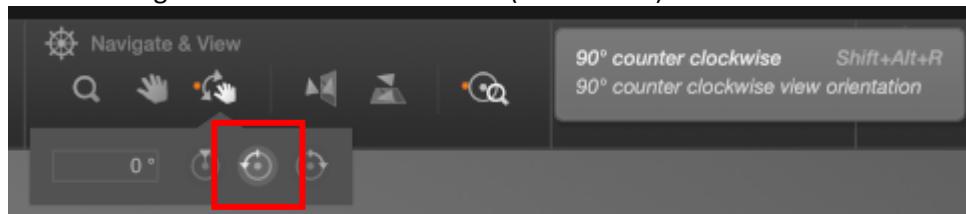
Inserire un valore numerico per la rotazione (gradi)



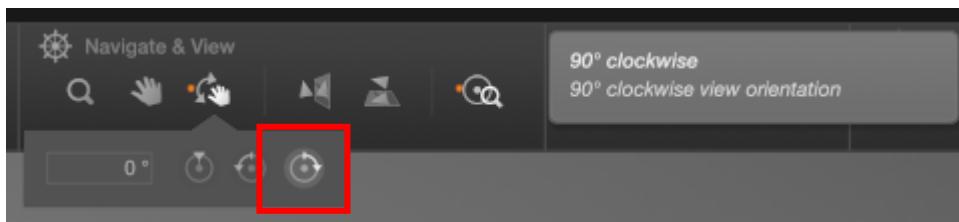
Ripristina la rotazione originale (**Ctrl+Shift+R**)



Ruota il disegno di 90° in senso antiorario (**Shift+Alt+R**)



Ruota il disegno di 90° in senso orario

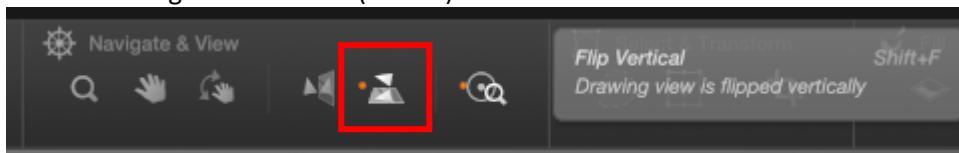


RIFLESSIONE

Riflette il disegno in orizzontale (**F**)

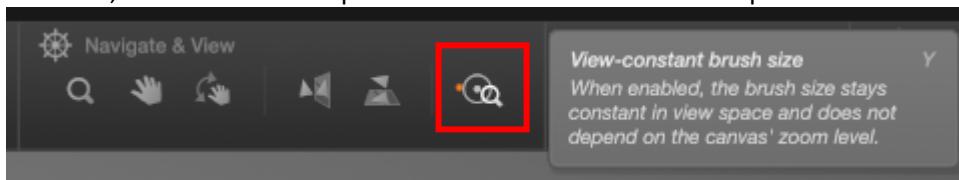


Riflette il disegno in verticale (**Shift+F**)



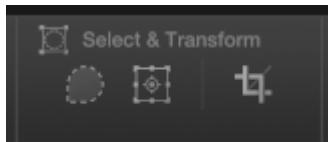
ZOOM e DIMENSIONE PENNELLO

Se attivo, la dimensione del pennello rimane costante e non dipende dal livello di zoom



Tutorial: Selezione e Trasformazione (Select & Transform)

Questa sezione controlla le operazioni di selezione e le relative trasformazioni:

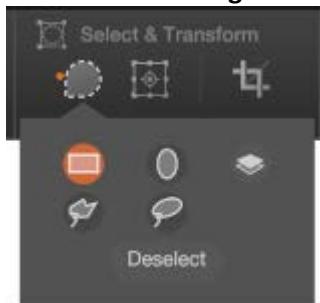


Lo strumento Selezione può essere attivato con il tasto “W”.

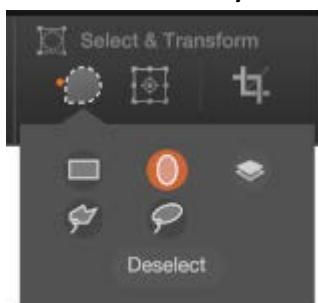
Quando lo strumento di selezione è attivo possiamo:

- 1) **Effettuare una normale selezione** con il mouse
- 2) **Aggiungere ad una selezione esistente** tenendo premuto il tasto “Shift”
- 3) **Sottrarre ad una selezione esistente** tenendo premuto il tasto “Alt”
- 4) **Intersecare con una selezione esistente**
- 5) **Modificare una selezione esistente**

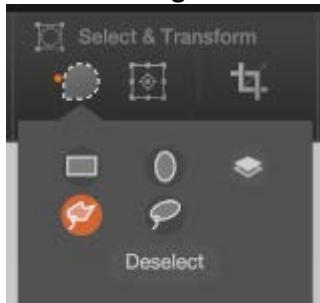
Selezione a Rettangolo



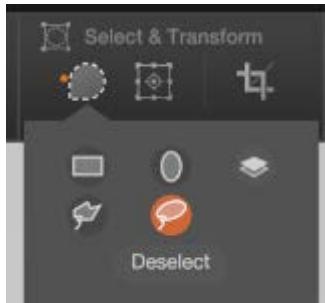
Selezione ad Ellisse/Cerchio(Shift)



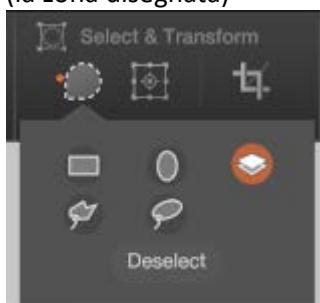
Selezione Poligonale



Selezione a Mano Libera



Seleziona il **canale Alfa**
(la zona disegnata)



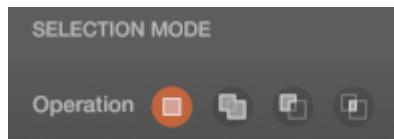
Deseleziona tutto
(Annulla Selezione)



Quando attiviamo lo strumento Selezione si apre il relativo pannello (a destra dell'area di lavoro):



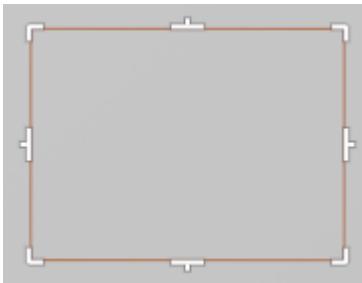
In questo pannello dobbiamo vedere le opzioni per la Modalità di Selezione (SELECTION MODE):



- Selezione di BASE
- Selezione in modalità UNIONE/ADDITIONE (premere Shift)
- Selezione in modalità SOTTRAZIONE (premere Alt)
- Selezione in modalità INTERSEZIONE

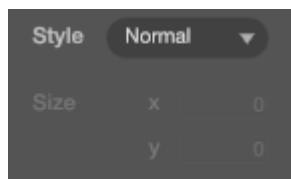
Questo ci permette di creare una selezione unica utilizzando più selezioni.

Una volta creata una selezione è possibile modificarla premendo il pulsante :

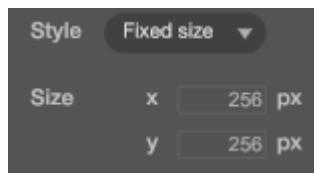


Adesso la selezione può essere ridimensionata e ruotata utilizzando le maniglie bianche posizionate intorno alla selezione (premere “**Enter**” per accettare o “**Esc**” per annullare).

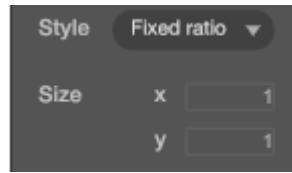
Infine possiamo “vincolare” il modo in cui tracciamo una Selezione nella seguente sezione:



Selezione senza vincoli



Selezione di grandezza fissa predefinita

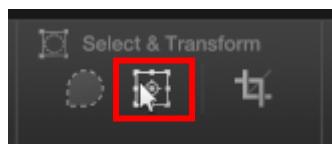


Selezione con rapporto x/y costante

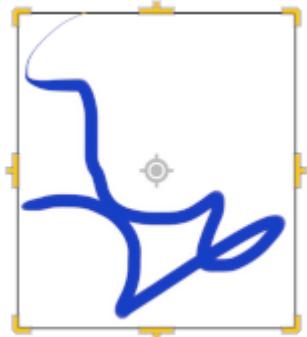
I tasti scorciatoia (shortcuts) per le selezioni sono i seguenti:

W	Attiva lo strumento Selezione
Shift / Alt	Aggiunge / Sottrae alla selezione attiva
Ctrl+A	Seleziona tutto
Ctrl+D	Deseleziona tutto (Annulla Selezione)
Ctrl+I	Inverte la Selezione
Shift+Ctrl+A	Seleziona il canale Alfa (la zona disegnata)

Adesso vediamo cosa si può fare con le **Trasformazioni**. Premendo il tasto evidenziato (oppure il tasto “**T**”):



Viene creato un rettangolo intorno alla zona selezionata oppure intorno a tutta l’area colorata del disegno:

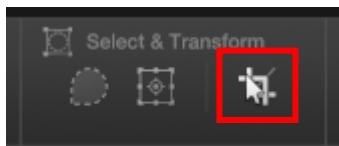


Questa zona può essere ridimensionata e ruotata utilizzando gli ancoraggi gialli posizionati intorno alla zona stessa e può essere spostata usando il bersaglio grigio al centro della zona. Provate da soli.

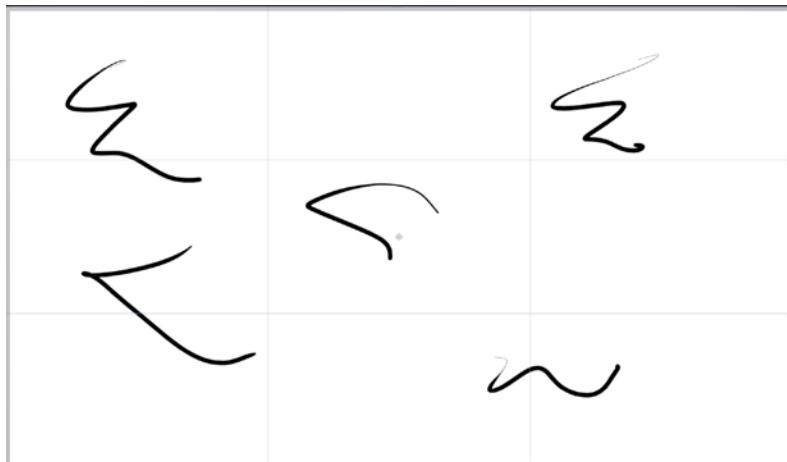
Tenendo premuto il tasto “Shift” la rotazione avviene per passi.

Quando abbiamo finito con le trasformazioni possiamo premere nuovamente il tasto per accettare le modifiche (anche con il tasto “Enter”) oppure premere il tasto “Esc” per annullare le modifiche.

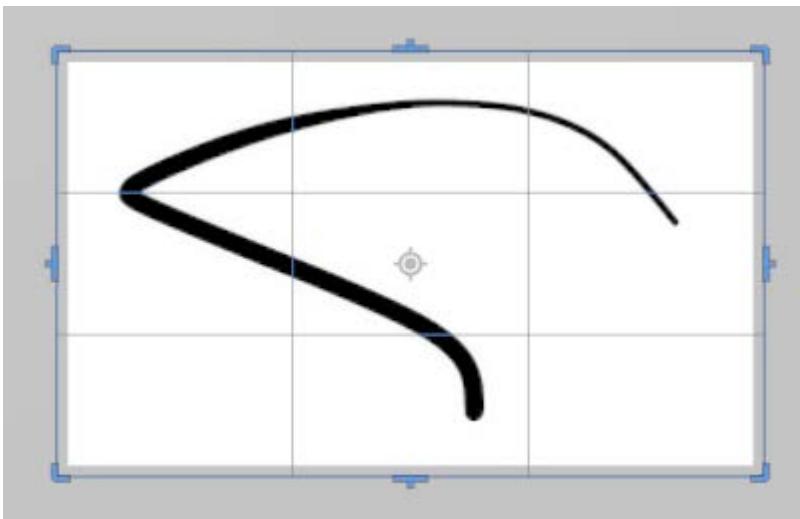
Infine abbiamo lo strumento **Ritaglia (Crop)** che ci permette di tagliare l’area di lavoro a nostro piacimento. Premendo il pulsante evidenziato oppure il tasto “Shift+C” (assicurandoci che non ci sia una selezione attiva):



Ottieniamo la seguente immagine (con alcune pennellate):



E possiamo definire con il mouse il rettangolo da ritagliare (colore azzurro):

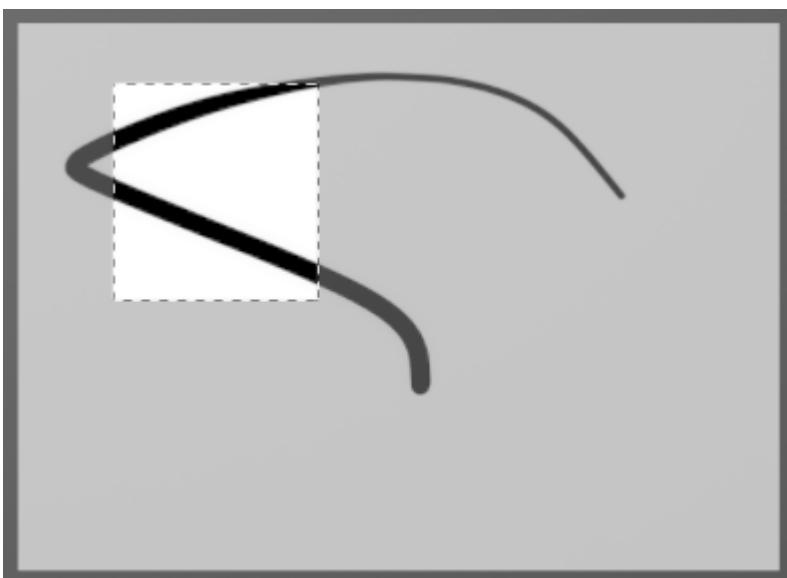


Prima di confermare il taglio possiamo ridimensionare o ruotare l'area del disegno definita dal rettangolo.
Provate da soli per vedere come si comporta il disegno.

Per accettare il taglio dell'area definita premere il tasto “**Enter**”.

Per annullare il taglio premere il tasto “**Esc**”.

NOTA: Se abbiamo una selezione attiva la pressione di questo pulsante provoca il taglio immediato di tutta e sola l'area della selezione:

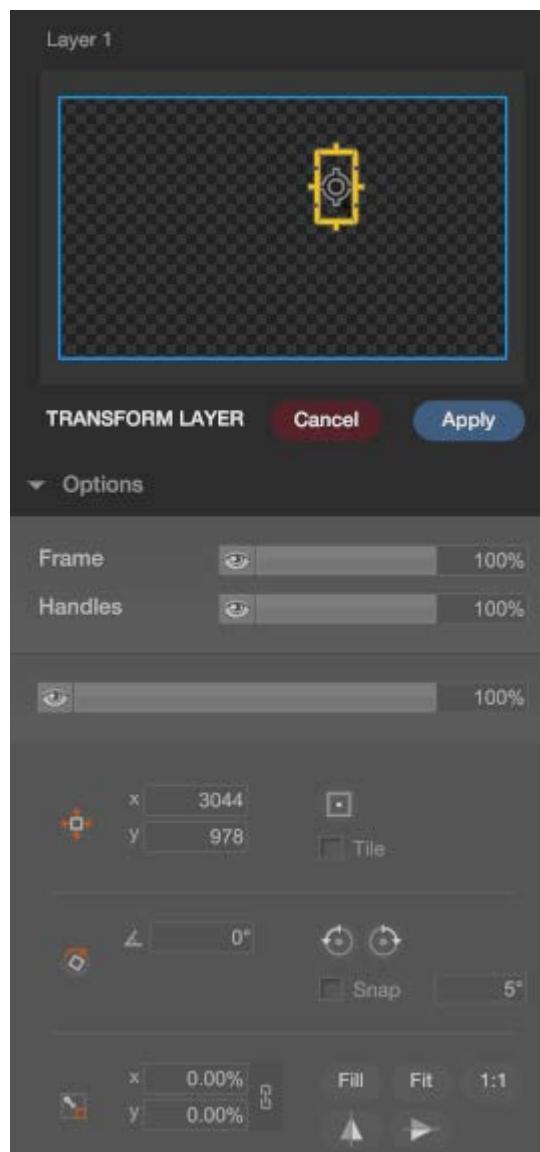


Con una selezione attiva premo il tasto **Ritaglia/Crop** (“**Shift+C**”)...



...e ritaglio l'area selezionata.

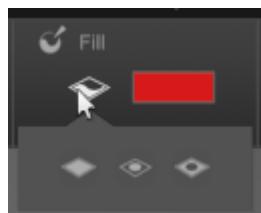
Quando attiviamo lo strumento Trasformazione ("T") si apre il relativo pannello (a destra dell'area di lavoro):



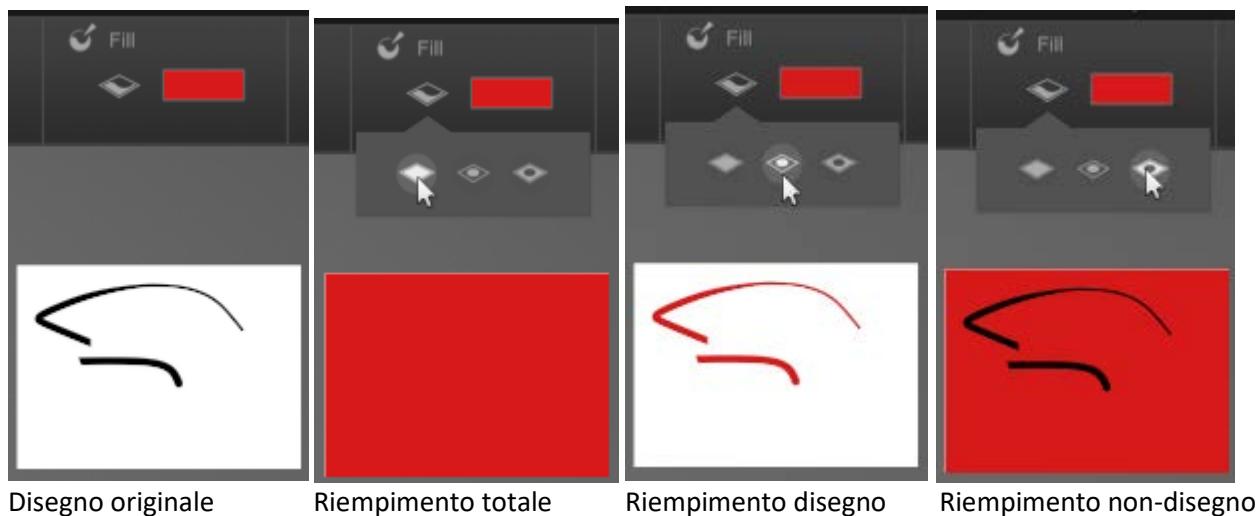
I parametri di questo pannello permettono una creazione più precisa della nostra Trasformazione.
È sufficiente fare qualche prova per capire come funzionano.

Tutorial: Riempimento (Fill)

Questo strumento controlla la modalità di riempimento della selezione e dell'area di lavoro.



Vediamo come funzionano i tre pulsanti con un esempio:



NOTA: Se abbiamo una selezione attiva viene riempita solo la zona selezionata.

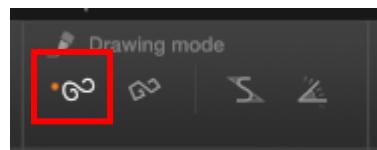
Il **rettangolo del colore** ha alcune funzionalità:



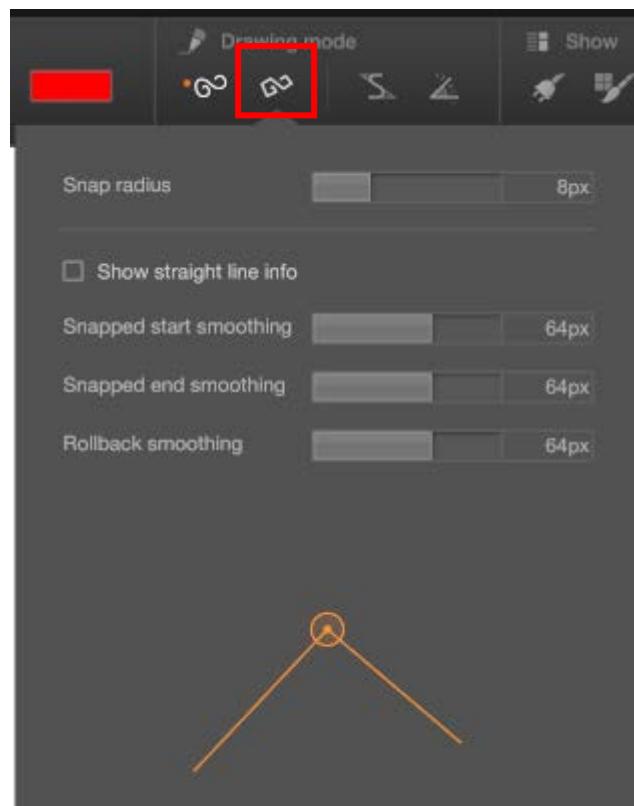
In alto troviamo la storia dei colori e in basso troviamo un cursore (slider) che ci permette di modificare l'opacità del colore selezionato.

Tutorial: Modalità di Disegno (Drawing Mode)

Questo pannello contiene alcuni parametri per il disegno assistito:

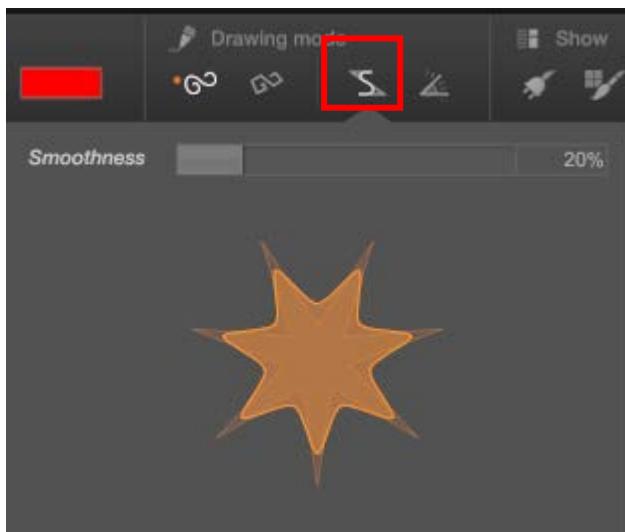


Disegno a Mano libera: disegno libero (nessun vincolo).

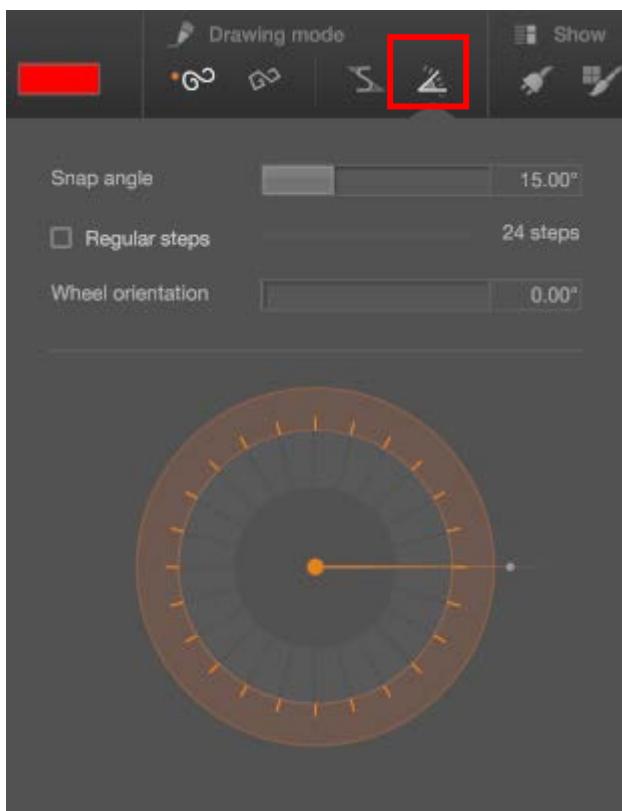


Disegno con linee rette: il disegno è vincolato a tracciare linee rette.

Possiamo definire il parametro di **Aggancio (Snap)** in pixel e le modalità di visualizzazione dei nodi di inizio e fine delle rette.

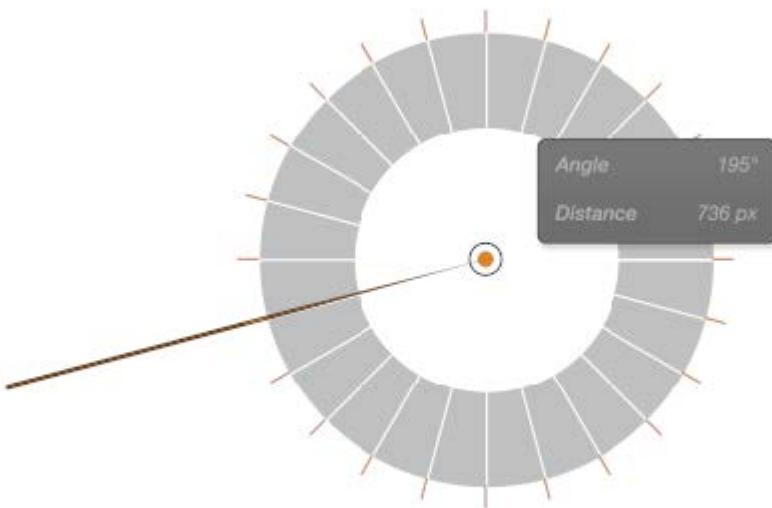


Morbidezza (Smoothness) del tratto (pennellata).



Impostazioni per il disegno con linee rette

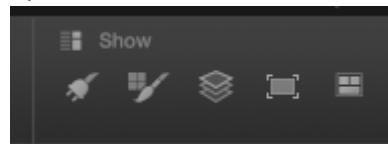
Quando siamo in modalità a mano libera (**Free Hand Mode**) possiamo disegnare con linee rette premendo il tasto il tasto “Shift”:



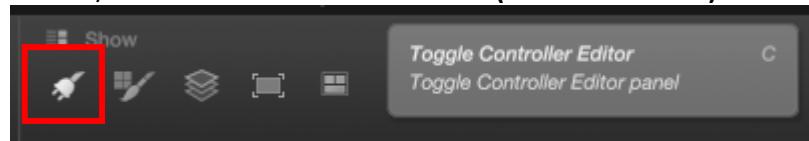
Provate da soli per scoprire tutte le varie possibilità.

Tutorial: Mostra Finestre/Pannelli (Show)

Questa sezione Permette di mostrare/nascondere le finestre dell'interfaccia grafica (UI – User Interface):



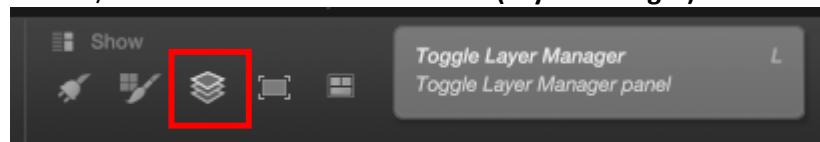
Mostra/Nasconde l'Editor dei Controller (**Controller editor**)



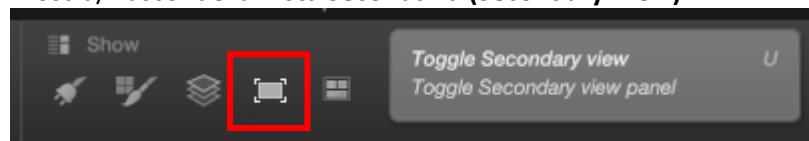
Mostra/Nasconde la Gestione dei Pennelli (**Brush Manager**) (floating)



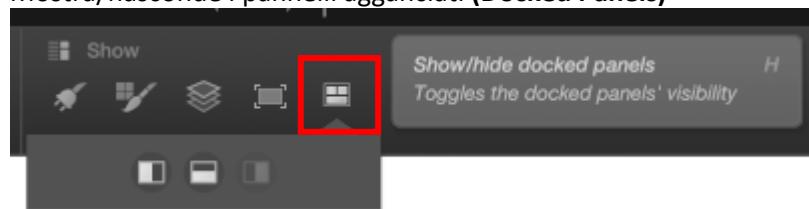
Mostra/Nasconde la **Gestione dei Livelli (Layer Manager)**

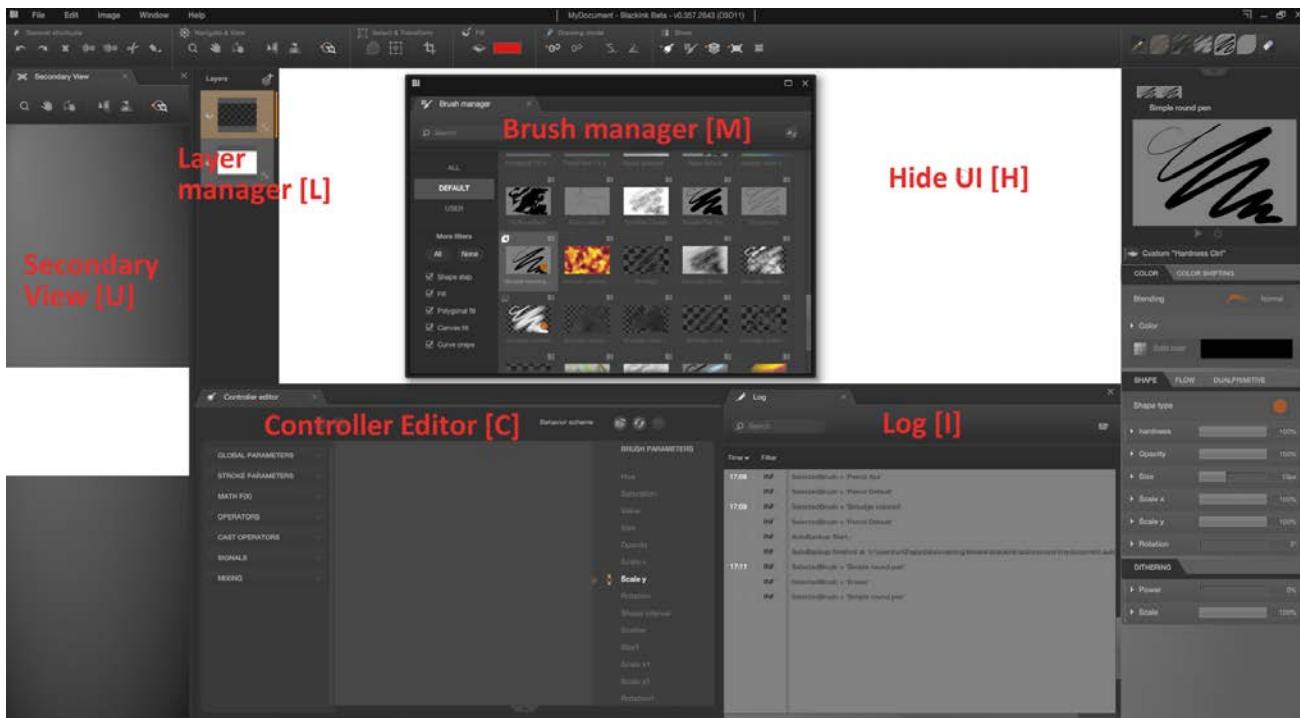


Mostra/Nasconde la **Vista Secondaria (Secondary View)**



Mostra/nasconde i pannelli agganciati (**Docked Panels**)

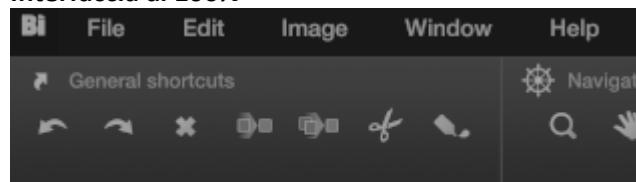




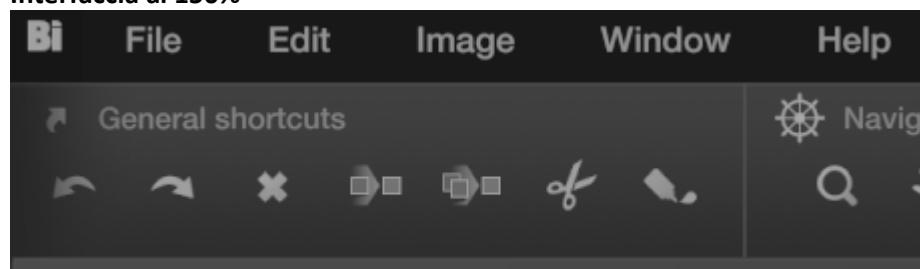
Tutorial: Zoom dell'Interfaccia Grafica

Possiamo ingrandire o ridurre le dimensioni dell'interfaccia istantaneamente utilizzando il tasto scorciatoia “Ctrl+Q” e spostando il mouse.

Interfaccia al 100%



Interfaccia al 150%



Fantastico!!!

Tutorial: Sostituzione dei pennelli predefiniti

In questo tutorial vedremo come cambiare i pennelli predefiniti dell'interfaccia.
Al primo avvio di BlackInk i pennelli predefiniti sono:

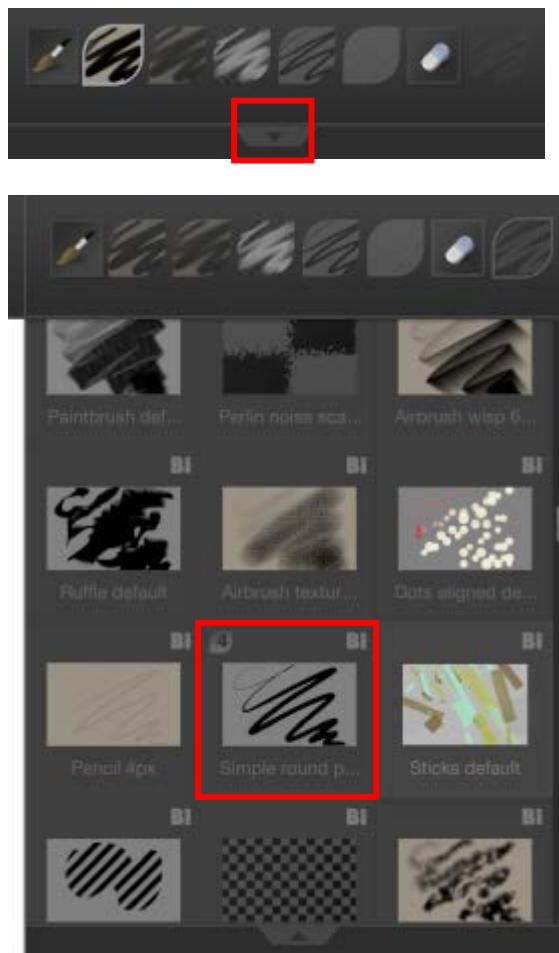
Felt pen 4px, Airbrush 12px, Smudge colored, Free fill, Angle line fill



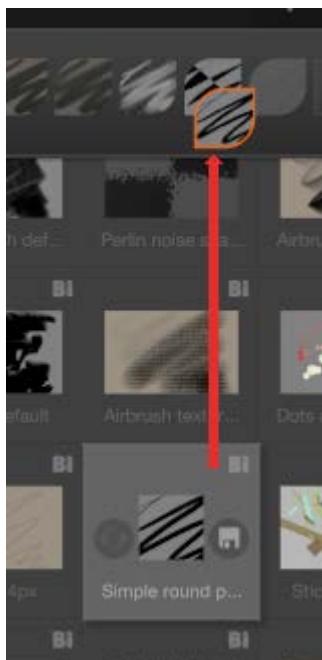
I pennelli sono cinque e possono essere selezionati usando i tasti 1..5 del tastierino numerico.

Supponiamo di voler cambiare questa impostazione e di inserire i pennelli **Simple round pen** e **Pencil Default** al posto di **Free fill** e **Angle line fill** (gli ultimi due).

Per fare questo occorre aprire la finestra **Brush Manager** premendo il tasto seguente:



Adesso dopo aver individuato il pennello (Simple round pen) occorre cliccarlo e trascinarlo (drag&drop) sopra il pennello che volete sostituire:

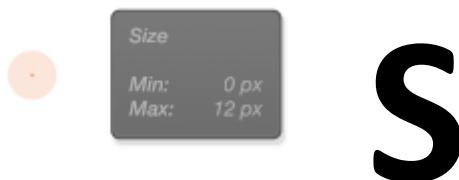


Questo è tutto: la prossima volta che aprirete BlackInk avrete la nuova impostazione di pennelli.

Tutorial: Modifica veloce della dimensione, dell'opacità e della morbidezza del pennello

Quando utilizziamo un pennello possiamo variare la sua **Dimensione (Size)**, la sua **Opacità (Opacity)** e la sua **Morbidezza (Smoothness)** senza utilizzare il relativo pannello dei parametri, ma con i tasti scorciatoia.

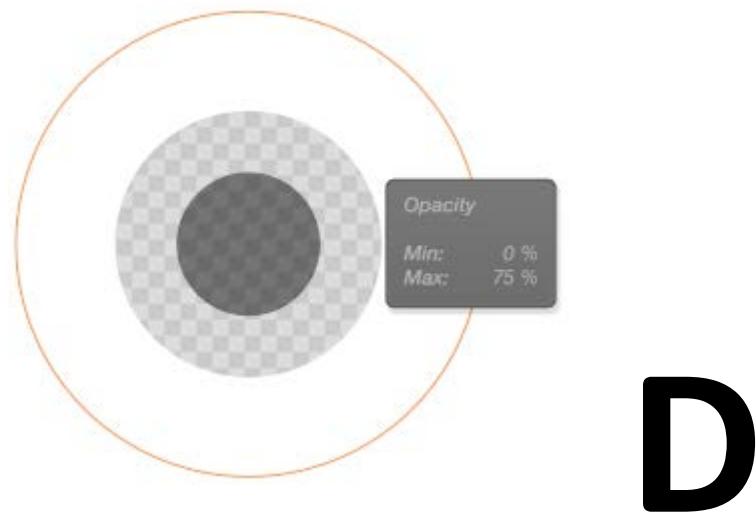
Per modificare la **Dimensione (Size)** occorre premere il tasto “**S**” per far comparire la seguente immagine:



Tenendo premuto il tasto “**S**” e muovendo il mouse **verso destra** la **dimensione aumenta**, mentre muovendo il mouse **verso sinistra** la **dimensione diminuisce**.

NOTA: Se la Dimensione minima è pari a 0 (zero), allora aumenta/diminuisce solo la Dimensione massima. Se la Dimensione minima è diversa da 0 (zero), allora aumentano/diminuiscono sia la Dimensione minima che la Dimensione massima (anche il loro intervallo cresce linearmente).

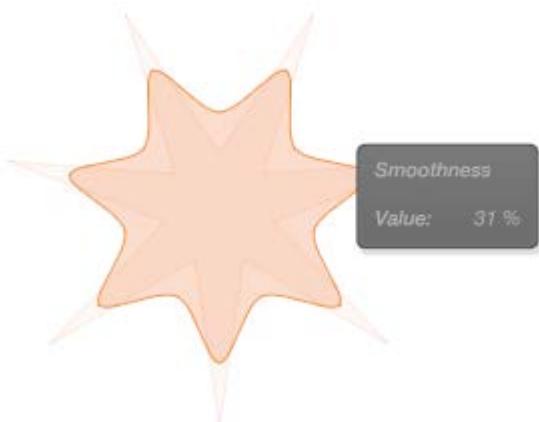
Per modificare l'**Opacità (Opacity)** occorre premere il tasto “**D**” per far comparire la seguente immagine:



Tenendo premuto il tasto “**D**” e muovendo il mouse **verso destra** l'**opacità aumenta**, mentre muovendo il mouse **verso sinistra** l'**opacità diminuisce**.

NOTA: Se l'Opacità minima è pari a 0 (zero), allora aumenta/diminuisce solo l'Opacità massima. Se l'Opacità minima è diversa da 0 (zero), allora aumentano/diminuiscono sia l'Opacità minima che l'Opacità massima (anche il loro intervallo cresce/diminuisce linearmente).

Per modificare la **Morbidezza (Smoothness)** occorre premere il tasto “**Z**” per far comparire la seguente immagine:



Z

Tenendo premuto il tasto “Z” e muovendo il mouse **verso destra la morbidezza aumenta**, mentre muovendo il mouse **verso sinistra la morbidezza diminuisce**.

Tutorial: Cambio veloce tra Pennello e Gomma (Eraser)

È possibile passare velocemente dal Pennello attivo alla Gomma (e viceversa) utilizzando dei tasti scorciatoia.

Per fare questo dobbiamo utilizzare alcuni tasti scorciatoia:

- 1) **Premendo e rilasciando il tasto “E”** passiamo alla Gomma (quella che è presente nel menu dei pennelli) e possiamo cancellare quello che vogliamo per poi ritornare al pennello attivo premendo il tasto “B” oppure nuovamente il tasto “E”.
- 2) **Premendo il tasto “E” (senza rilasciarlo)** possiamo cancellare parti del disegno utilizzando come gomma il pennello attivo. Al termine delle operazioni di cancellazioni **rilasciamo il tasto “E”** e ritorniamo automaticamente ad avere il pennello attivo come strumento di disegno.

In questo modo le operazioni di disegno e di cancellazione sono molto più fluide.

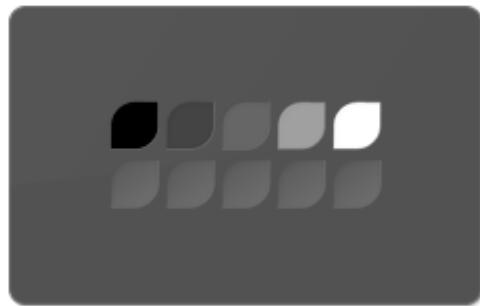
Tutorial: Cattura dei colori dal disegno

Possiamo selezionare un colore del disegno tenendo premendo il tasto “Alt” e cliccando sul disegno:



Questo è l'immagine del cursore quando è attivo lo strumento **Cattura Colore (Pick Color)**. Il cerchio colorato rappresenta il colore attivo (rosso), mentre la zona esterna rappresenta il colore che si sta catturando in quella posizione (bianco).

Tutorial: Colori preferiti



Puoi sempre avere accesso ai colori che hai definito (colori solidi, gradienti (sfumati) e basati su immagini).

Avvio di BlackInk

I colori impostati come preferiti verranno automaticamente caricati all'avvio dell'applicazione.

Salvataggio

Trascina e rilascia il colore desiderato su uno slot vuoto per salvarlo.

Copia

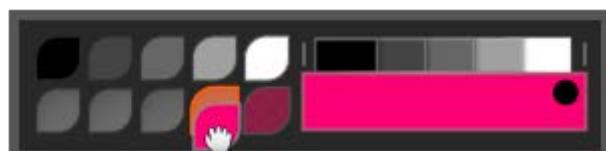
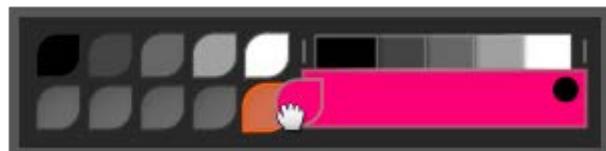
Per copiare un colore, trascinalo e rilascia un altro slot.

Sostituzione

Trascina e rilascia il tuo colore in un colore esistente.

Eliminazione

Per eliminare il tuo colore, prendilo e trascinalo fuori dalla finestra dei preferiti e quando appare l'icona del cestino, rilascialo.



Tutorial: I Cursori (Slider)

Sono dispositivi di scorrimento che permettono di modificare il valore di un parametro (Pressione, Opacità).

Cursori semplici

Il semplice cursore **modifica il valore massimo** che il parametro può raggiungere durante la pennellata.



È possibile **modificare manualmente il valore massimo del cursore facendo clic sul valore e digitando un numero oppure trascinando orizzontalmente**:



Nota: l'opacità non può essere impostata a più del 100%

Cursori Estesi

Fai clic sulla freccia accanto al nome del parametro per aprire una versione estesa del cursore:



Nella versione estesa è possibile **modificare i valori di minimo e di massimo**:



Il valore minimo è il valore più basso che il parametro può avere durante il tratto del pennello:



In questo esempio, la forma non sarà mai più piccola di 7 pixel quando la pressione è bassa e mai più grande di 23 pixel quando si preme più forte sullo stilo:



Puoi **modificare entrambi i valori minimo e massimo allo stesso tempo** spostando il piccolo blocco sopra il cursore:



Questo **modificherà i due valori mantenendo la loro differenza relativa**.

Impostando lo stesso valore su minimo e massimo si disabilita il controller e il parametro avrà sempre lo stesso valore (es. la dimensione non cambierà rispetto alla pressione dello stilo):



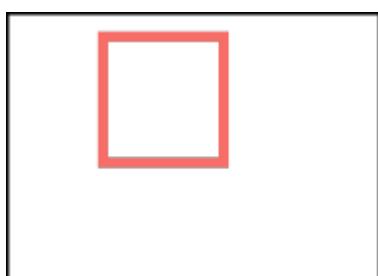
Tutorial: I Livelli



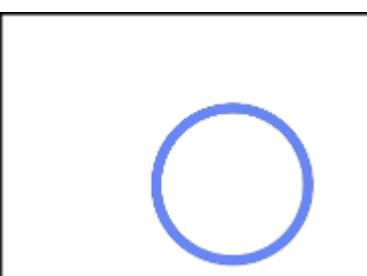
Cosa sono i livelli

Immaginate di avere dei fogli lucidi, ad esempio quelli che vengono usati per proiettare delle slide)

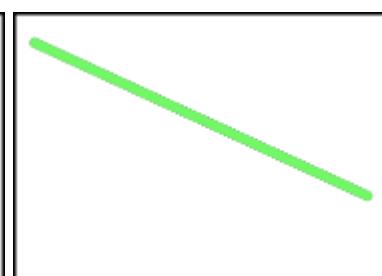
Su ogni foglio possiamo disegnare qualcosa: sul primo disegniamo un quadrato, sul secondo un cerchio e sul terzo una linea:



Foglio 1 (Livello 1)

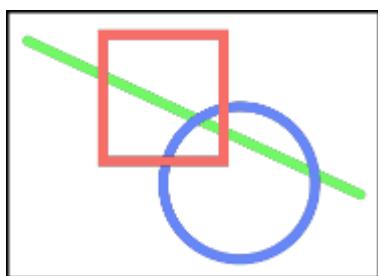


Foglio 2 (Livello 2)



Foglio 3 (Livello 3)

Cosa succede se sovrapponiamo i tre lucidi? Avremo un'unica immagine, data dalla somma dei tre fogli sovrapposti:



Fogli sovrapposti

I livelli permettono di fare la stessa cosa (e anche di più) con il nostro disegno.

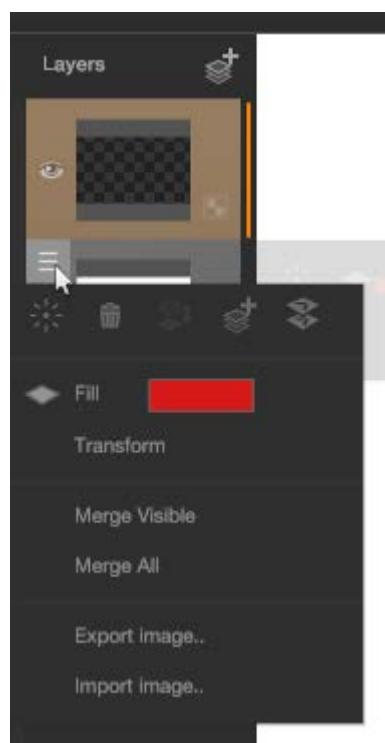
Alcune considerazioni ed operazioni sui livelli:

- Quando un livello è disegnato completamente rende invisibili i livelli sottostanti (a meno che a questo livello non venga applicata una certa trasparenza).
- Quando disegniamo possiamo modificare solo il livello attivo (gli altri livelli non vengono modificati).
- Possiamo applicare un valore di trasparenza a tutto il livello.

- Possiamo riordinare i livelli a nostro piacimento.
- Possiamo rendere totalmente invisibile un livello.
- Possiamo cancellare totalmente tutto quello che abbiamo disegnato su un livello.
- Possiamo duplicare un livello
- Possiamo unire (fondere) due livelli in uno solo.

Con BlackInk possiamo avere **otto livelli** diversi.

Possiamo visualizzare/nascondere la finestra dei livelli con il tasto “L”.



I pulsanti hanno le seguenti funzionalità:

- | | |
|--|---|
| | Crea un nuovo livello (Create new layer) |
| | Mostra/Nasconde livello (Show/Hide layer) |
| | Preserva trasparenza (Preserve transparency). È possibile disegnare solo sulle parti colorate. |
| | Pulisce il livello (Clear layer) |



Riempie il livello con il colore di Riempimento di default (**Fill with default color**)



Elimina il livello (**Delete layer**)



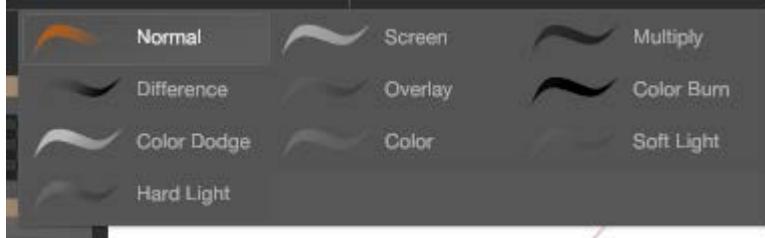
Unisce il livello con il livello sottostante (**Merge Down**)



Duplica il livello (**Duplicate layer**)



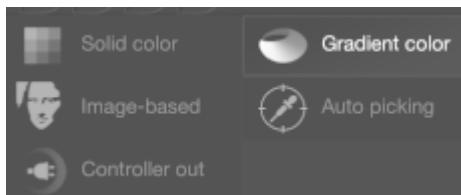
Modalità di fusione livello (Layer blending)



Valore di opacità/trasparenza del livello (**Layer opacity**)

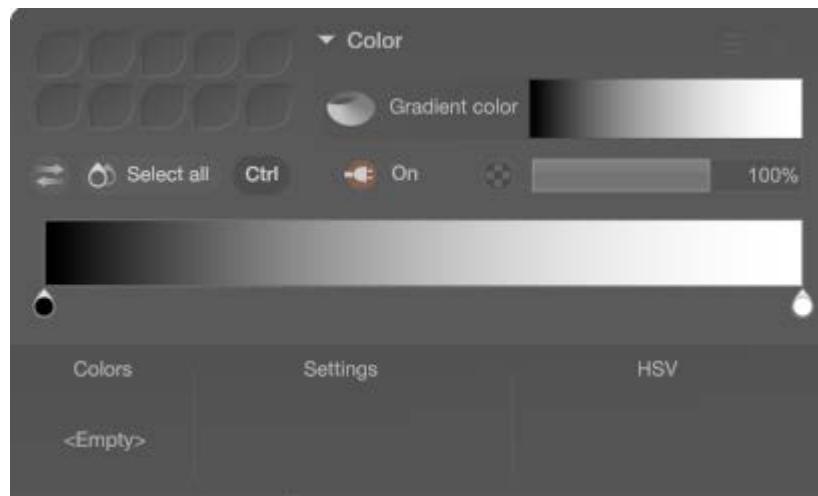
Tutorial: I Gradienti

Come già notato la scheda colori di BlackInk offre cinque modalità colore: **Solid color**, **Gradient color**, **Image-based**, **Auto picking**, **Controller out**. Questo tutorial riguarderà il **Gradient color**.

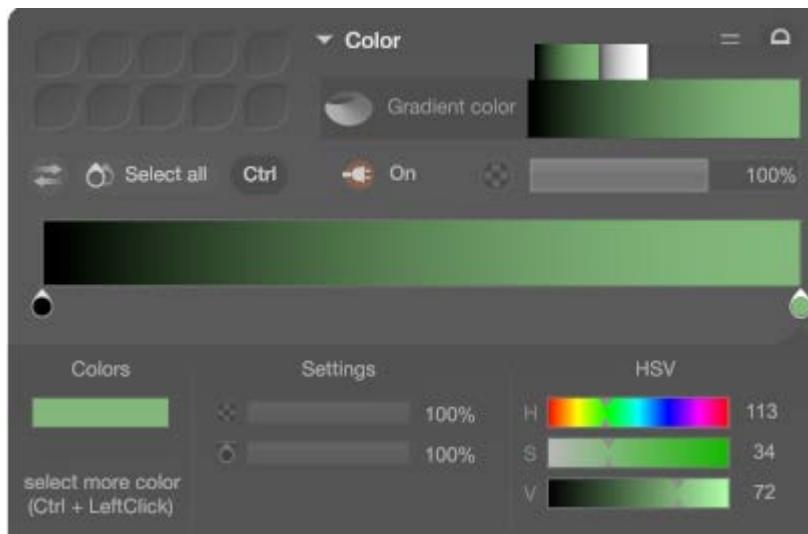


I pennelli predefiniti di BlackInk possono utilizzare i gradienti. Per default il colore seguirà la pendenza in funzione della pressione dello stilo, ad eccezione dei pennelli legati alla velocità del tratto. Questo, naturalmente, può essere cambiato con i controller, ma lo faremo in un altro tutorial. Per adesso vediamo come definire un gradiente.

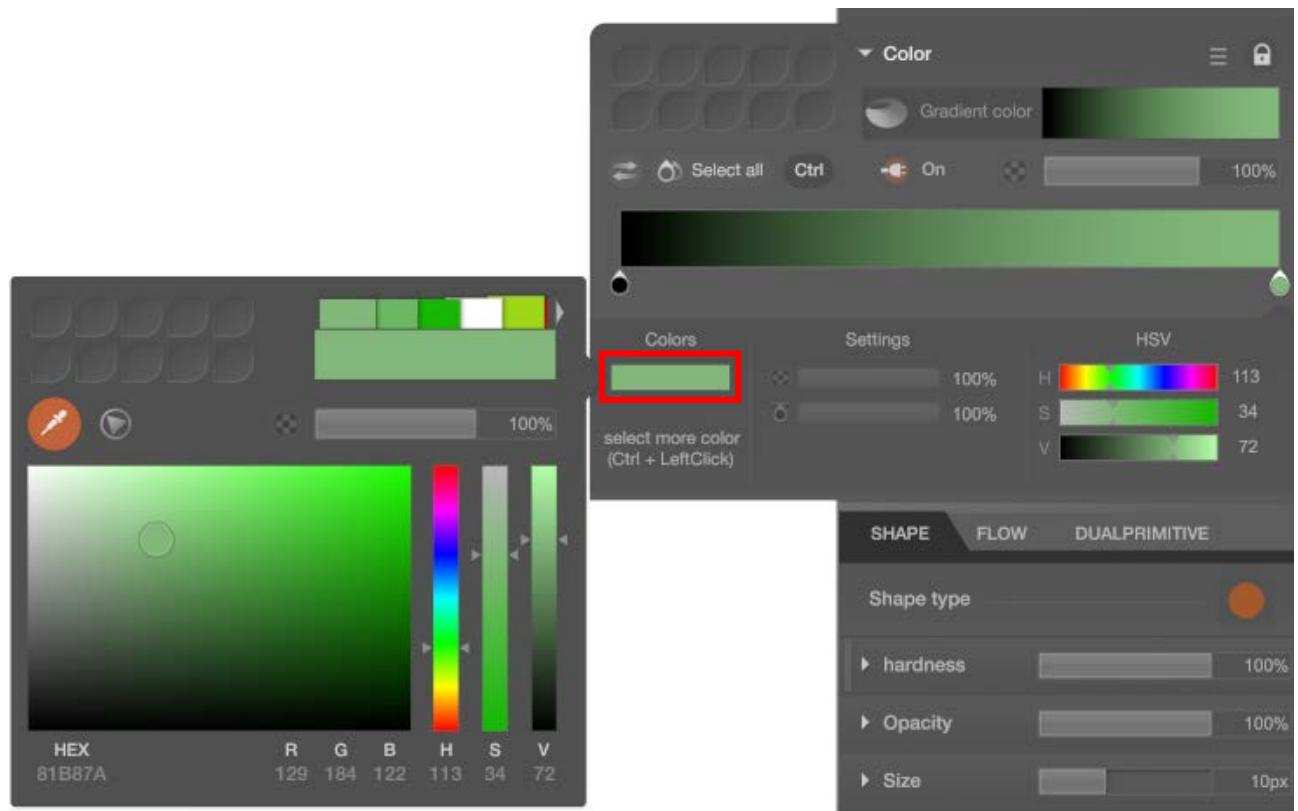
Quando si seleziona il **Gradient Color**, si vedrà il cambiamento di colore di anteprima da una tinta unita ad un gradiente. E quando si fa clic su questa piccola anteprima, appare l'editor del gradiente.



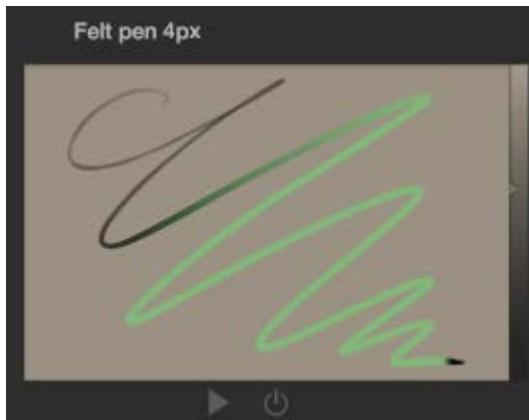
Le sfumature sono definite con "campioni". Questi sono simili a piccole gocce sotto la rampa gradiente. Ognuno di questi campioni definisce un colore. Selezionandone uno, vedrete alcune opzioni:



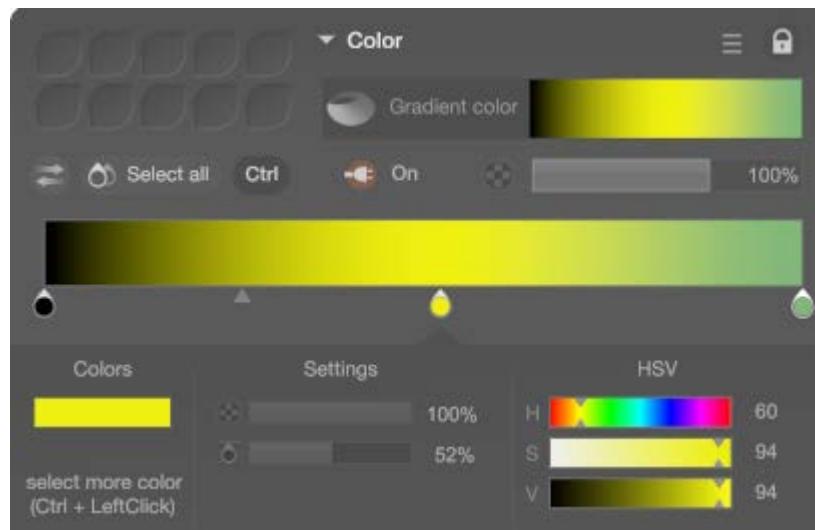
Quando si fa clic sulla piccola anteprima a colori a sinistra, si aprirà l'editor di colore standard. È lo stesso del semplice tinta unita con i suoi favoriti, la storia dei Colori e il Contagocce (Pick color). Questo è il principale strumento per la definizione dei colori nel gradiente.



E, naturalmente, l'anteprima pennello ci mostrerà come il colore viene applicato al pennello finale:



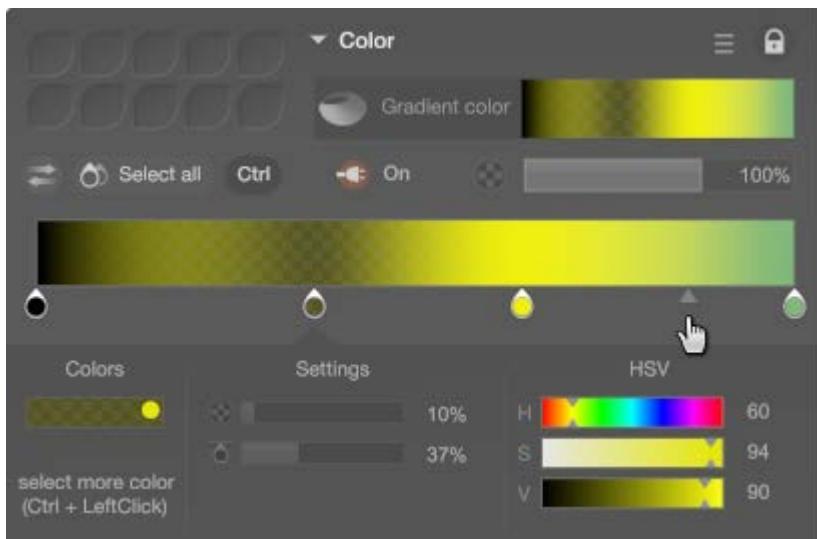
Torniamo ai campioni. È possibile spostarli come si desidera e aggiungerne di più cliccando appena sotto la rampa di pendenza principale. Per eliminare un campione, basta trascinarlo fuori dalla zona campioni (in alto o in basso).



Si noti che anche il valore di **Alfa** del colore del campione si rifletterà nel gradiente.

Si possono avere parti trasparenti e opache nello stesso gradiente.

Inoltre è possibile modificare la linearità del gradiente tra due campioni spostando la freccia grigia indicata dal cursore:



Ora vediamo le altre opzioni nella parte inferiore (che sono relative al campione attivo):



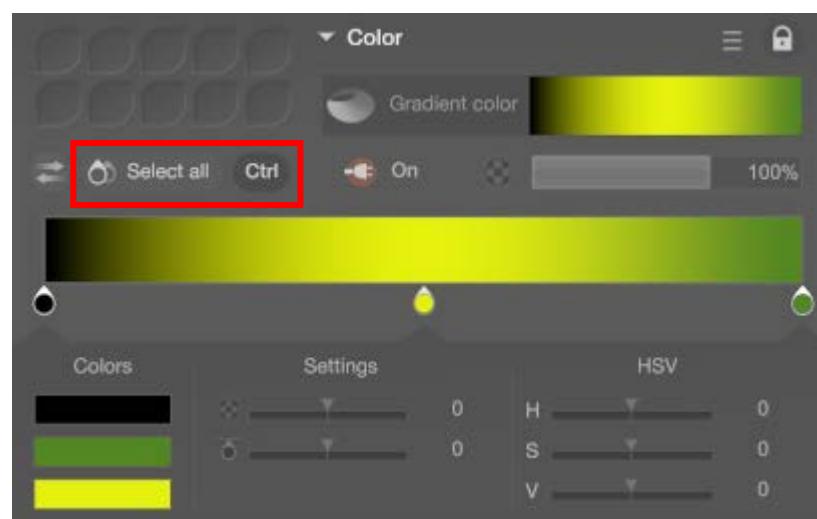
In primo luogo, abbiamo un campo "**Impostazioni**" (**Settings**). Il primo cursore modifica l'alfa del vostro colore (uguale ad aprire l'editor dei colori e modificare il valore di Alfa).

Il cursore in basso cambia la posizione del campione, proprio come quando ci si sposta direttamente con il mouse, ma qui è anche possibile inserire un numero esatto.

Poi nella zona "**HSV**", è possibile modificare la tonalità, la saturazione e il valore del colore (uguale ad aprire l'editor dei colori e modificare il colore).

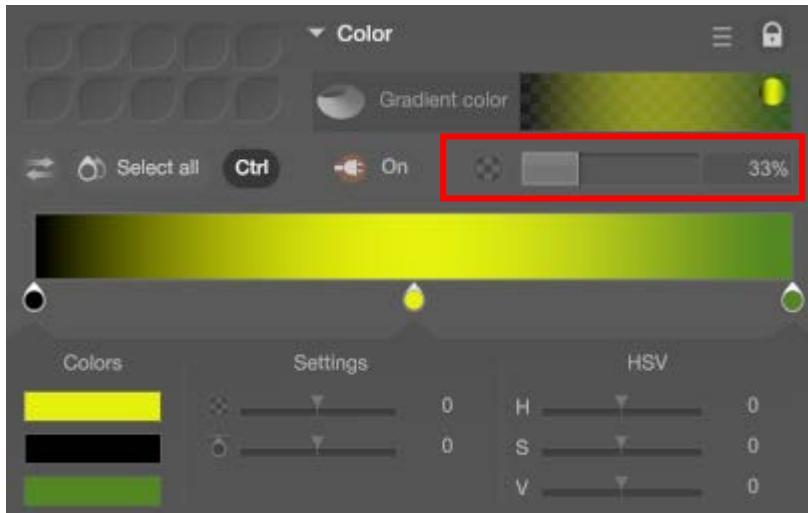
Inoltre osserviamo che è possibile selezionare più campioni!

Per scegliere i campioni a uno a uno, attivare il pulsante "**ctrl**" al di sopra del gradiente oppure tenere premuto il tasto "**Ctrl**" sulla tastiera. Per selezionare tutti i campioni, utilizzare il pulsante "Selezione tutto" ("**Select All**"):



Si può vedere nella parte inferiore a sinistra una lista di tutti i colori selezionati. E ora, utilizzando le "Impostazioni" (Settings) e i cursori "HSV", è possibile modificarli contemporaneamente! È possibile modificare i valori di tutti gli alfa, spostare tutti a sinistra o a destra, saturare ogni colore, e così via. Queste modifiche si aggiungono alle impostazioni iniziali dei colori.

Sopra la rampa gradiente principale è un altro, cursore alfa più grande:



Questo cambierà l'opacità della sfumatura finale, ma non modifica i colori selezionati.
Se si desidera solo una pendenza costante trasparente, è possibile utilizzare questo dispositivo di scorrimento invece che modificare singolarmente ogni colore.
Ma la maggior parte del tempo si può decidere di lasciare l'opacità del pennello al 100%.

Per finire vediamo gli ultimi due pulsanti della finestra gradiente:

- Inverte la posizione dei campioni del gradiente attivo
- On permette di gestire il gradiente con un controller (lo vedremo meglio in un altro tutorial)

E questo è tutto, è possibile utilizzare i gradienti ogni volta che vuoi.
Essi sono un ottimo strumento per il disegno dei volumi, senza mai cambiare colore, è sufficiente premere più o meno sul tablet:



Possono aggiungere piacevoli variazioni e un volume variabile:



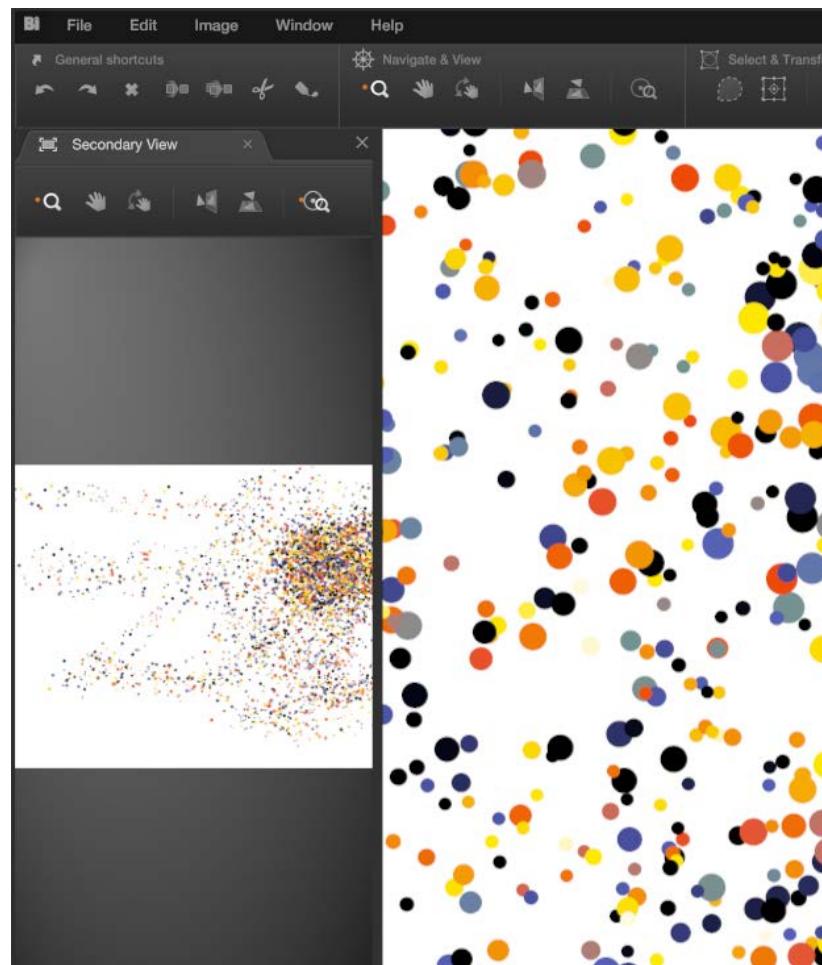
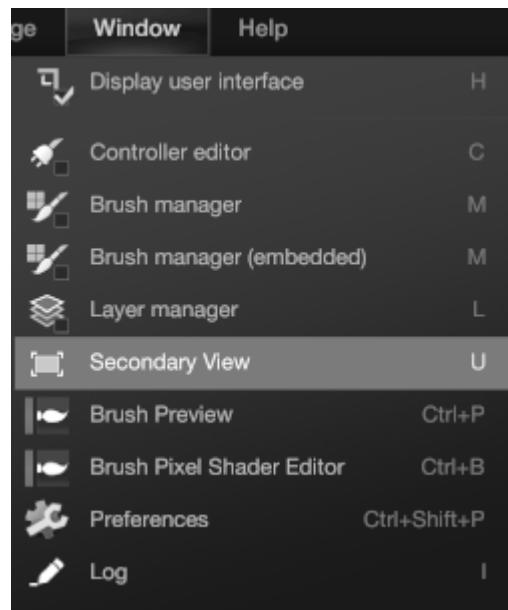
E, naturalmente, creare effetti psichedelici folli:



Divertiti!

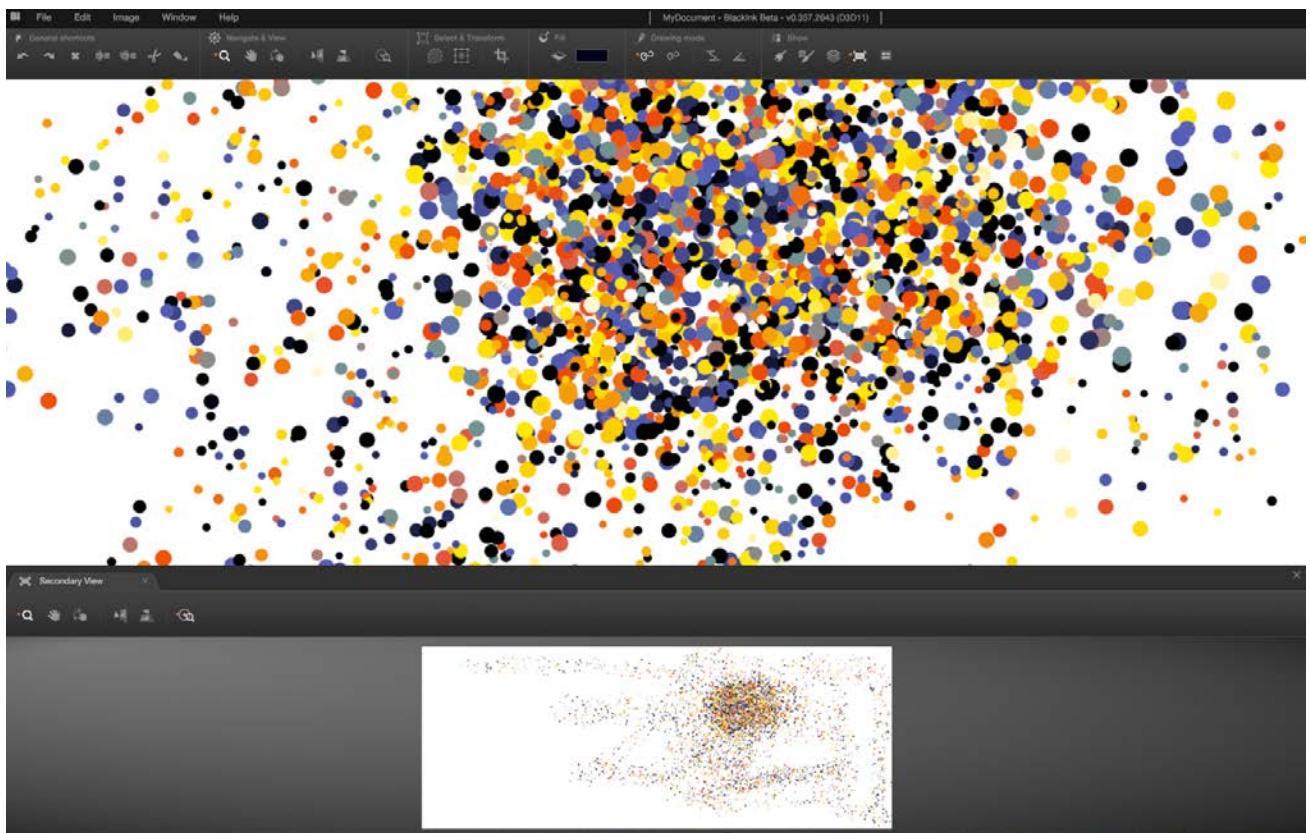
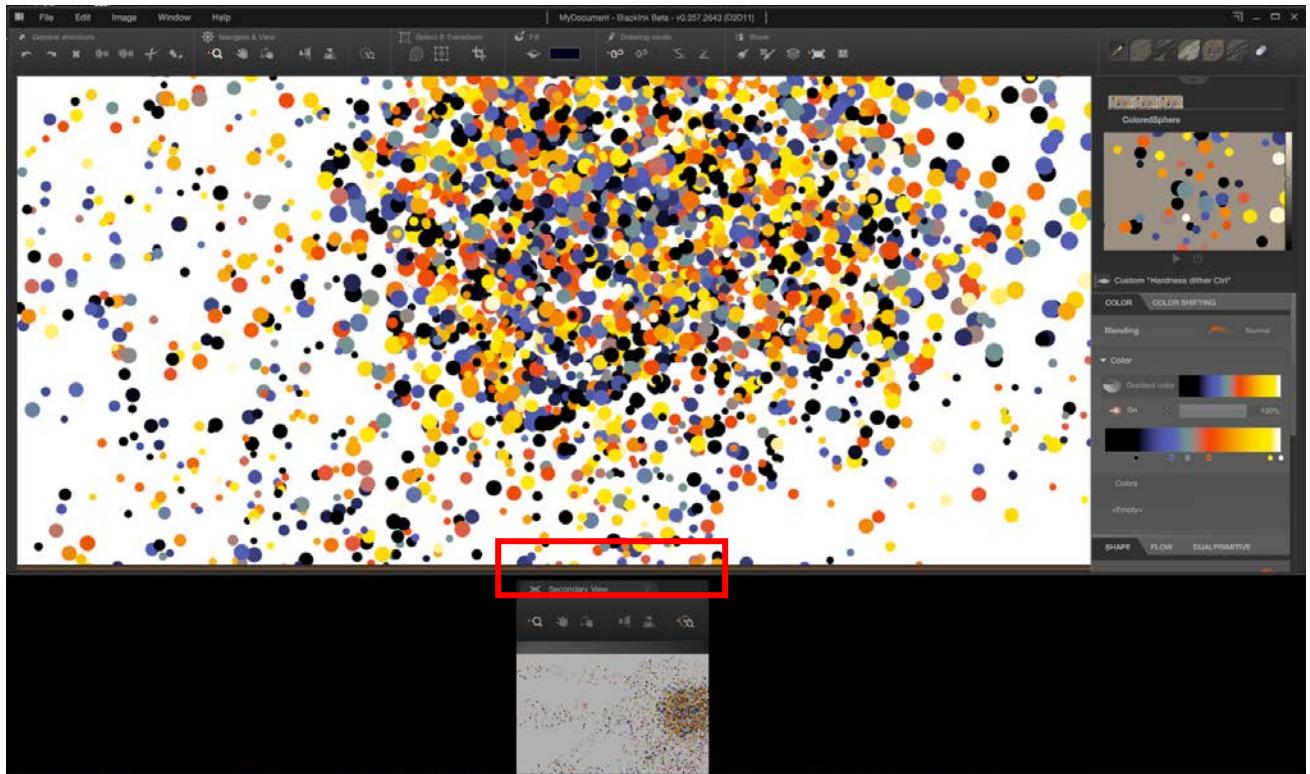
Tutorial: La finestra secondaria (Secondary View)

La finestra della **Vista Secondaria** si apre con il menu **Windows->Secondary View** o con il tasto “U”:



Questa vista è molto comoda perché ci consente di visualizzare il nostro disegno ad un'altra scala mentre lavoriamo sull'area principale. All'inizio è situata a sinistra dell'interfaccia, ma è possibile posizionarla in qualunque parte si desidera.

Per agganciarla in basso, clicciamo e trasciniamo la finestra fino a che non compare una **linea arancione** e a quel punto rilasciamo la finestra:



Se abbiamo un secondo monitor possiamo spostare questa finestra e aprirla a schermo intero:

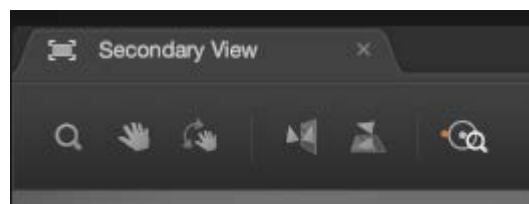


Monitor principale: Interfaccia BlackLink



Monitor secondario: Secondary View

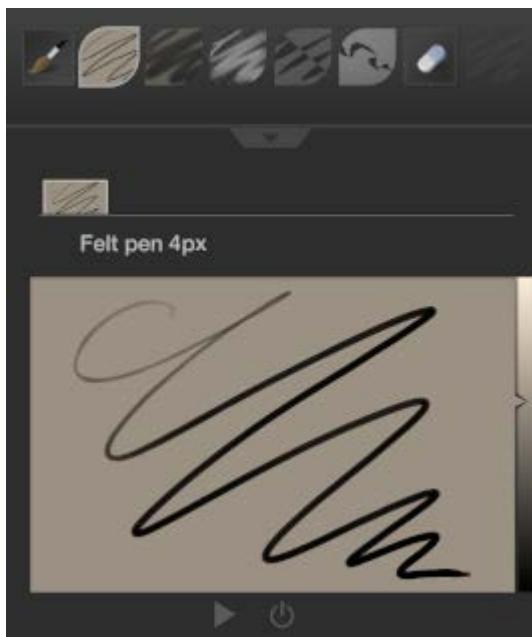
La Secondary View può essere gestita con i comandi standard (Zoom, Sposta, Ruota, Capovolgi) dal menu:



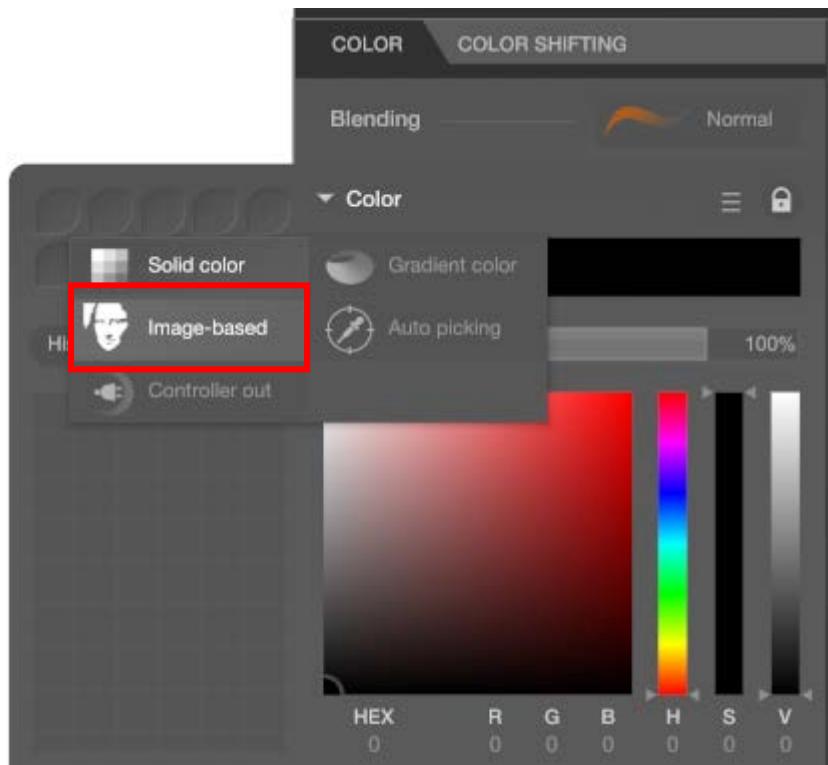
NOTA: Il pennello attivo funziona anche se disegniamo direttamente sulla Vista Secondaria.

Tutorial: Creazione di un disegno da un'immagine (color ImageBased)

Selezioniamo il pennello (brush) **Felt pen 4px**:

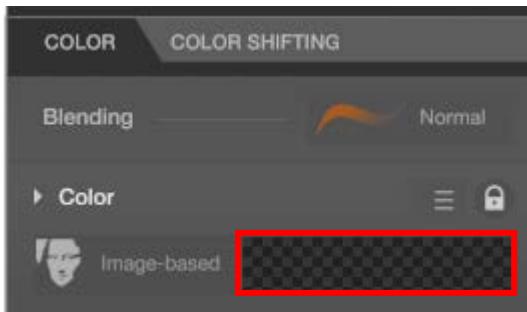


Adesso selezioniamo **Color->Image-Based**:

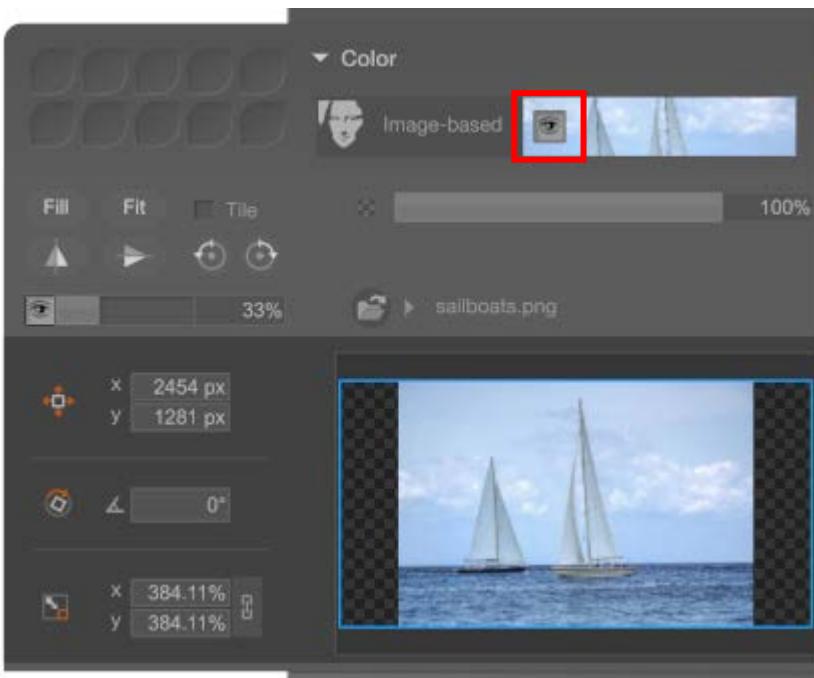


In questo modo il colore utilizzato per disegnare verrà preso dall'immagine di riferimento.

Per caricare la nostra immagine dobbiamo cliccare sull'area trasparente:

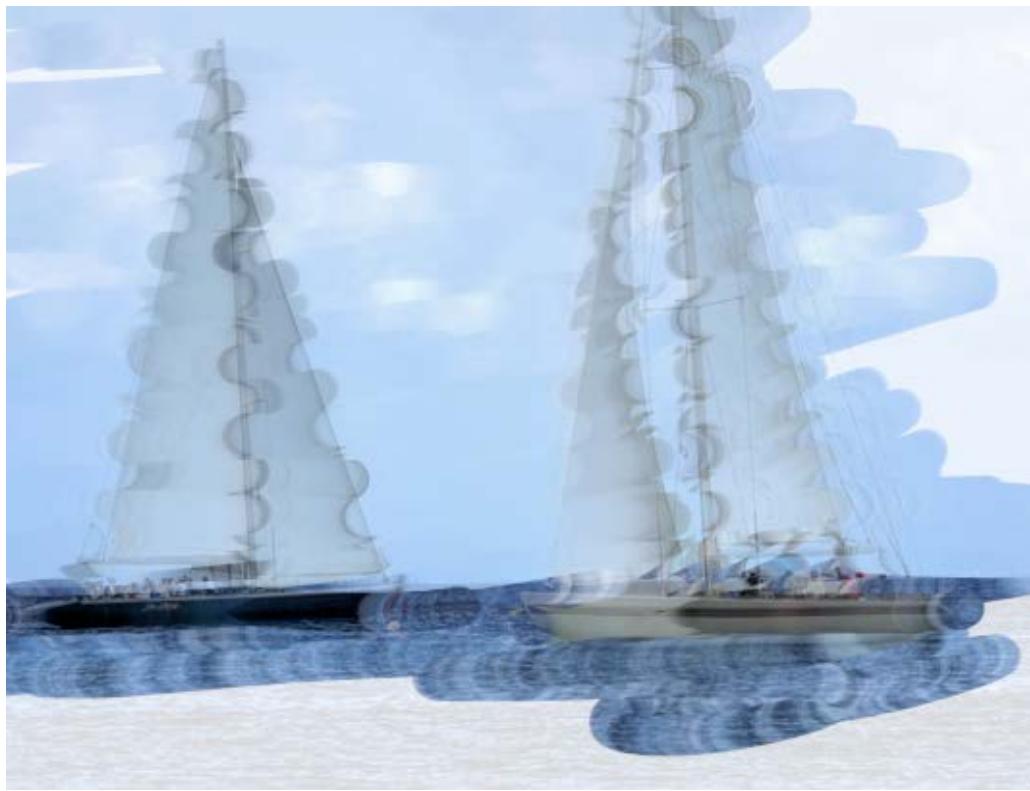


Dopo aver scelto l'immagine da caricare abbiamo la seguente situazione:



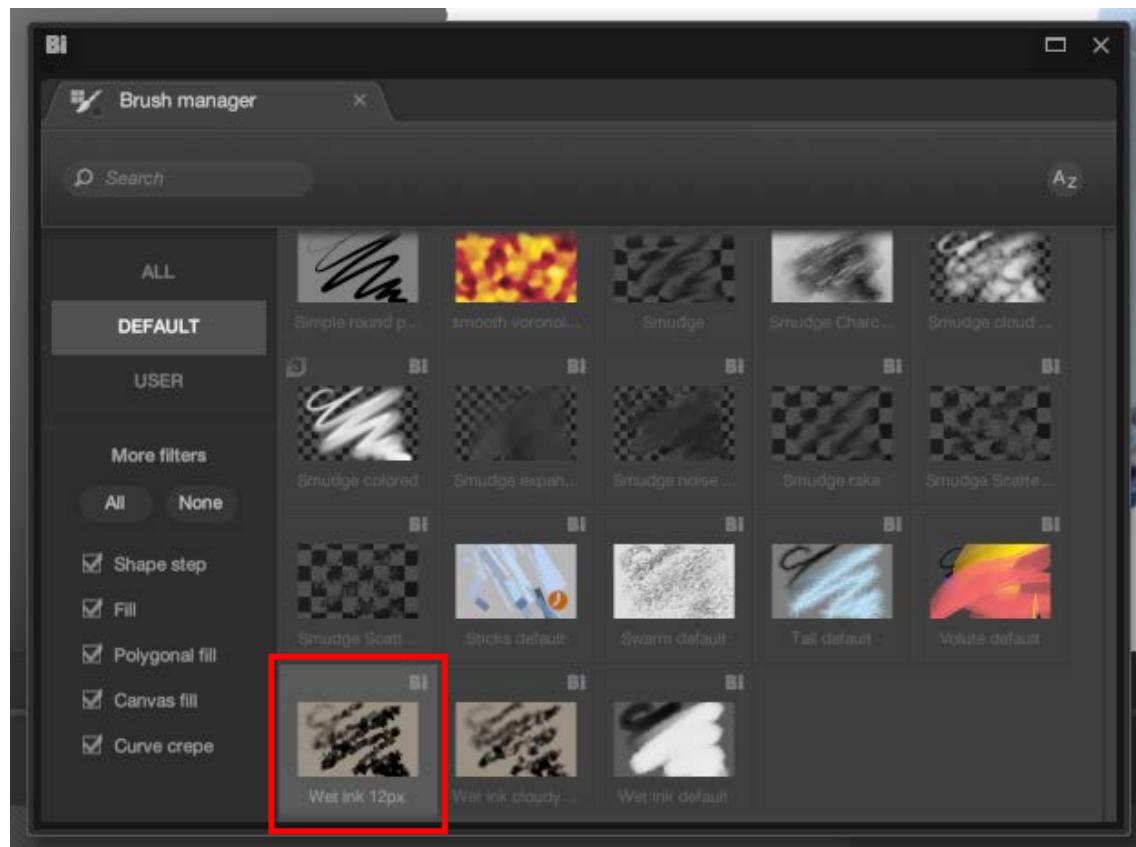
Questa finestra ci permette di scalare e posizionare l'immagine nel disegno a nostro piacimento (provate da soli a modificare i vari parametri). Possiamo anche modificare la trasparenza dell'immagine. L'icona a forma di Occhio permettere di attivare/disattivare la visualizzazione dell'immagine

Siamo pronti: se disegniamo sopra l'immagine otteniamo un risultato simile al seguente:

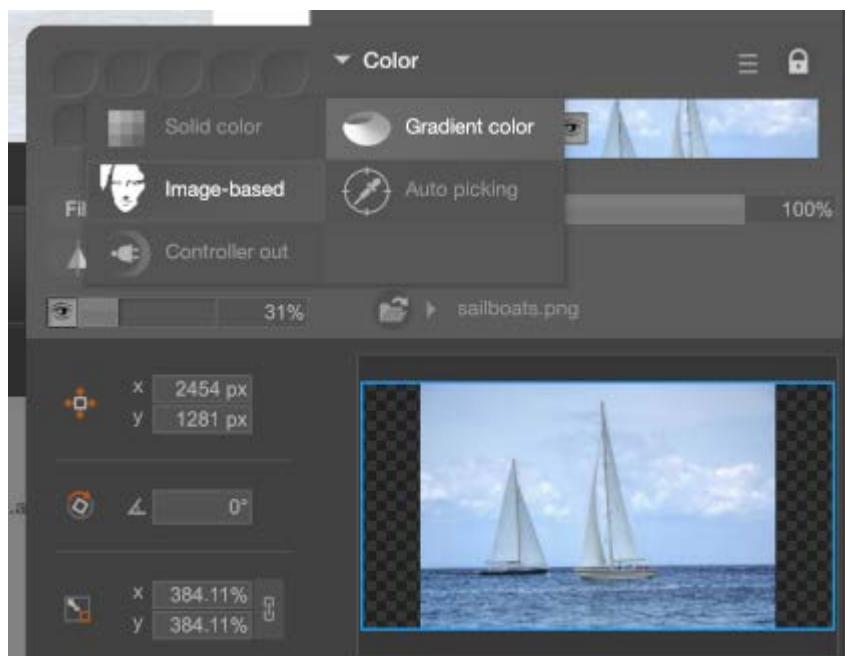


Per disegnare possiamo utilizzare qualunque tipo di pennello (basta assicurarsi che non abbia nessun controller applicato al colore).

Prendiamo ad esempio il pennello Wet ink 12px. Premendo il tasto "M" compare la finestra di gestione Pennelli (Brush Manager):



Cambiamo il colore (COLOR) a Imagebased:

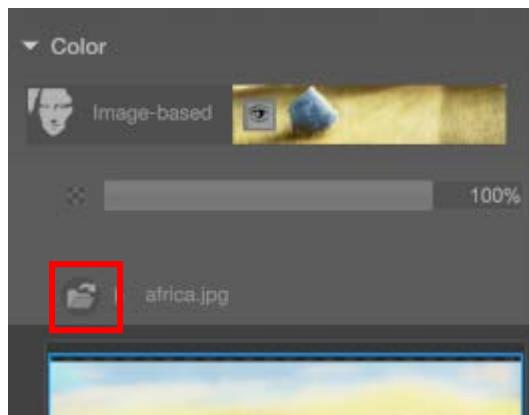


Disegniamo e otteniamo:



Sicuramente voi farete di meglio.

Se volete caricare un'altra immagine dovete premere il seguente tasto:



Tutorial: La struttura dei file dei Colori Preferiti

BlackInk salva automaticamente i **Colori Preferiti** in tre file di testo diversi (uno per ogni modalità di colore) nella cartella:

c:\Users\<username>\AppData\Roaming\Bleank\BlackInk\UserColors

Per la modalità **Solid color** il file si chiama **Current.CSolid**.

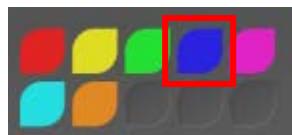
Per la modalità **Gradient color** il file si chiama **Current.CGradient**.

Per la modalità **Image-based** il file si chiama **Current.CImage**.

Apriamo i file con un editor di testo e vediamo come sono strutturati:

Current.CSolid

```
<BlackInkColorSolid>
    <c slot="0" rgba="(0.74572,0.0167413,0.0167413,1)" />
    <c slot="1" rgba="(0.74572,0.713964,0.0167413,1)" />
    <c slot="2" rgba="(0.0167413,0.74572,0.034478,1)" />
    <c slot="3" rgba="(0.0235526,0.0167413,0.74572,1)" />
    <c slot="4" rgba="(0.74572,0.0167413,0.546302,1)" />
    <c slot="5" rgba="(0.0167413,0.737819,0.74572,1)" />
    <c slot="6" rgba="(0.74572,0.250285,0.0167413,1)" />
</BlackInkColorSolid>
```



Dove:

slot="3" indica in quale slot si trova il colore (parte dal numero zero che si trova in alto a sinistra).

rgba="(0.0235526,0.0167413,0.74572,1)" indica il colore con i valori di Red, Green, Blue e Alfa (compresi tra 0 e 1).

Current.CGradient

```
<BlackInkColorGradient>
  <c slot="0">
    <stops>
      <s p="0.60441" m="0.5" color="(0.00093549,0.00165396,0.0102626,1)"/>
      <s p="0.916903" m="0.5" color="(0.0259277,0.0259277,0.0259277,1)"/>
      <s p="1" m="0.5" color="(0.166829,0.166829,0.166829,1)"/>
    </stops>
  </c>
  <c slot="1">
    <stops>
      <s p="0.18" m="0.5" color="(0,0,0,1)"/>
      <s p="0.38" m="0.5" color="(0.0915184,0.113921,0.481952,1)"/>
      <s p="0.47" m="0.5" color="(0.170138,0.288816,0.288816,1)"/>
      <s p="0.59" m="0.5" color="(0.859174,0.0677246,0.00196342,1)"/>
      <s p="0.94" m="0.5" color="(1,0.827726,0,1)"/>
      <s p="1" m="0.5" color="(1,1,1,1)"/>
    </stops>
  </c>
  <c slot="2">
    <stops>
      <s p="0.18" m="0.5" color="(0,0,0,1)"/>
      <s p="0.38" m="0.5" color="(0.0915184,0.113921,0.481952,1)"/>
      <s p="0.47" m="0.5" color="(0.170138,0.288816,0.288816,1)"/>
      <s p="0.59" m="0.5" color="(0.859174,0.0677246,0.00196342,1)"/>
      <s p="0.94" m="0.5" color="(1,0.827726,0,1)"/>
      <s p="1" m="0.5" color="(1,1,1,0)"/>
    </stops>
  </c>
</BlackInkColorGradient>
```



Dove:

slot="1" indica in quale slot si trova il gradiente.

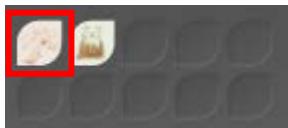
p="0.47" indica la posizione del campione di colore (il valore 0 (zero) si trova a sinistra e il valore 1 (uno) si trova a destra).

m="0.5" indica la posizione della freccia grigia rispetto al prossimo campione.

color="(0.170138,0.288816,0.288816,1)" indica il colore con i valori di Red, Green, Blue e Alfa (i valori sono compresi tra 0 e 1).

Current.CImage

```
<BlackInkColorImage>
  <c slot="0">
    <image path="c:\documents\img\demo.png"/>
  </c>
  <c slot="1">
    <image path="c:\documents\img\start.png"/>
  </c>
</BlackInkColorImage>
```



Dove:

slot="0" indica in quale slot si trova l'immagine gradiente.

image path="c:\documents\img\demo.png" indica il percorso completo dove è memorizzata l'immagine.

Possiamo creare questi file manualmente con i nostri colori preferiti oppure crearli con BlackInk e poi rinominarli per avere la possibilità di caricarli successivamente.

Sul web potete trovare le palette dei grandi artisti al sito: <http://www.colorlisa.com/>

Ecco una paletta con alcuni colori del quadro “Notte stellata” di Van Gogh (**Current.CSolid**):

```
<BlackInkColorSolid>
  <c slot="0" rgba="(0.0278473,0.015274,0.0125656,1)"/>
  <c slot="1" rgba="(0.543945,0.370605,0,1)"/>
  <c slot="2" rgba="(0.513672,0.525391,0.176514,1)"/>
  <c slot="3" rgba="(0.0702515,0.166626,0.419678,1)"/>
  <c slot="4" rgba="(0.015686,0.0470581,0.270508,1)"/>
</BlackInkColorSolid>
```



Tutorial: Le cartelle utente di BlackInk

Durante l'installazione BlackInk crea alcune cartelle destinate a salvare i file creati dall'utente (per esempio i file dei pennelli personalizzati).

Vediamo dove sono e cosa contengono. Aprite una finestra DOS (**cmd.exe**), spostatevi nella cartella utente di BlackInk (**cd “\Users\<username>\AppData\Roaming\Bleank\BlackInk”**) ed eseguite il comando “**dir**”:

Directory of

```
C:\Users\<username>\AppData\Roaming\Bleank\BlackInk
<DIR> AutoRecover
<DIR> Cache
<DIR> DebugProfiling
<DIR> Textures
<DIR> UserBrushes
<DIR> userbsl
<DIR> UserColors
<DIR> UserControllers
<DIR> userdatas
<DIR> UserLayout
    AppParams.xml
    BlackInk.Log
    blackink0-swap.bktmp
    Config.xml
    DebugParams.bkdbg
    Markers.mkrs
    MiniDump.dmp
```

Vediamo le cartelle più importanti:

UserBrushes	Contiene i file dei pennelli creati dall'utente (.bkbrush)
Userbsl	Contiene i file degli shader BSL creati dall'utente (.bsl)
UserColors	Contiene i colori preferiti creati dall'utente (.CSolid, .CGradient, .CImage)
UserControllers	Contiene i controllers creati dall'utente (.bkctrl)
Textures	Contiene le seguenti sottocartelle: Background Brushes Paper Pattern

La cartella **Texture** è quella che si apre di default quando importiamo una texture da BlackInk; le sottocartelle (**Background**, **Brushes**, **Paper**, **Pattern**) contengono i seguenti file:

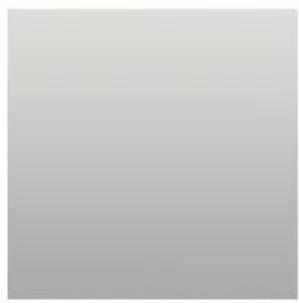
Background



BrushPreview_BKG00_Voronoi...



BrushPreview_BKG01_Voronoi...



BrushPreview_BKG02_GridLight



BrushPreview_BKG03_GridGrey



BrushPreview_BKG04_GridDark



BrushPreview_BKG04_GridDia...



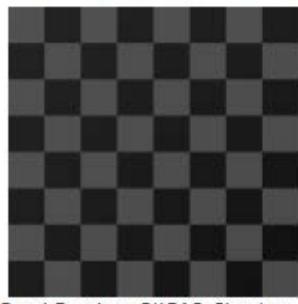
BrushPreview_BKG05_GridRai...



BrushPreview_BKG16_Checker...



BrushPreview_BKG16_Checker...



BrushPreview_BKG16_Checker...



BrushPreview_BKG16_Checker...



BrushPreview_BKG20_Voronoi...



BrushPreview_BKG20_Voronoi...



BrushPreview_BKG20_Voronoi...



BrushPreview_BKG20_Voronoi...



BrushPreview_BKG20_Voronoi...



BrushPreview_BKG20_Voronoi...



BrushPreview_BKG20_Voronoi...

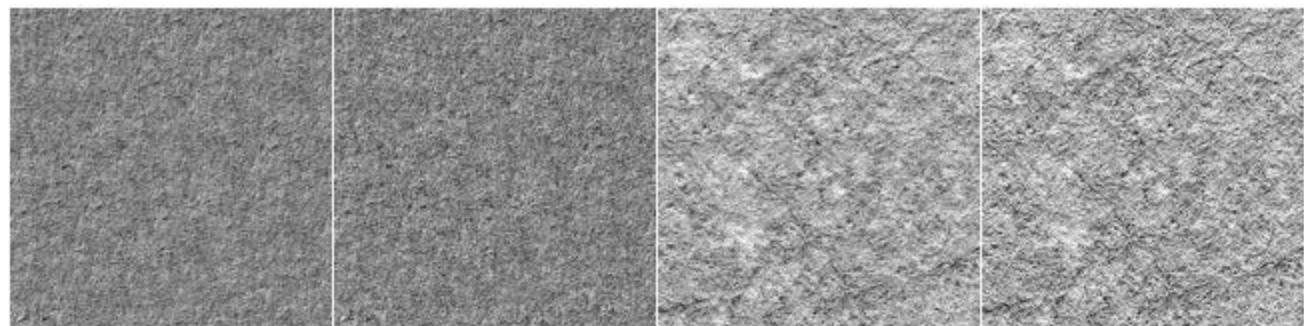


BrushPreview_BKG20_Voronoi...

Brushes



Paper

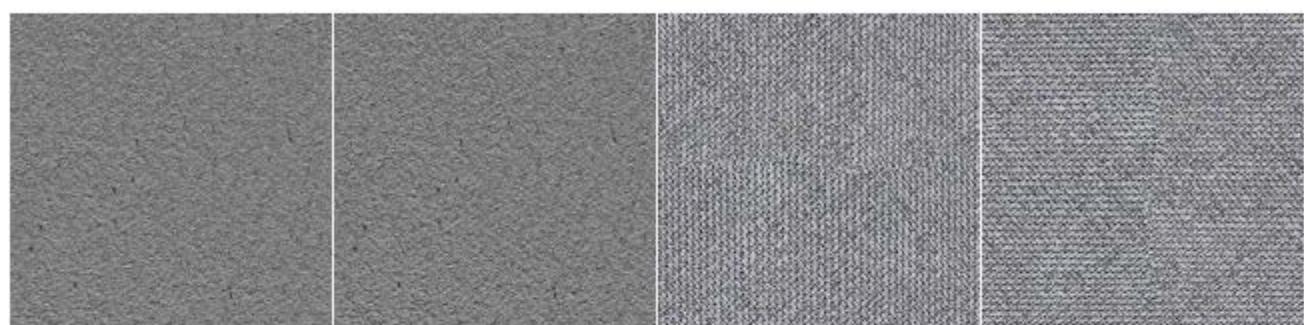


Paper_01

Paper_01_small

Paper_02

Paper_02_small

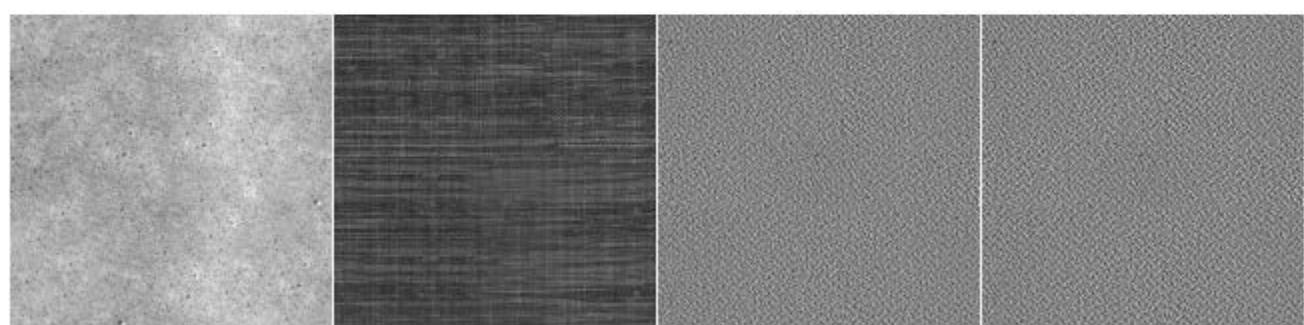


Paper_03

Paper_03_small

Paper_04

Paper_04_r90



Paper_05

Paper_06

Paper_07

Paper_07_small



Paper_09

Paper_09_small

Paper_10

Paper_10_small

Pattern



Nature_01

Nature_02

Potete salvare le vostre immagini (texture, background, ecc.) nelle cartelle che vi sembrano più opportune, in questo modo potete avere i vostri file preferiti a portata di mano.

Brush workshop 0: Introduzione ai controller

Questa è un'introduzione ad uno degli strumenti più potenti i BlackInk: i Controller.

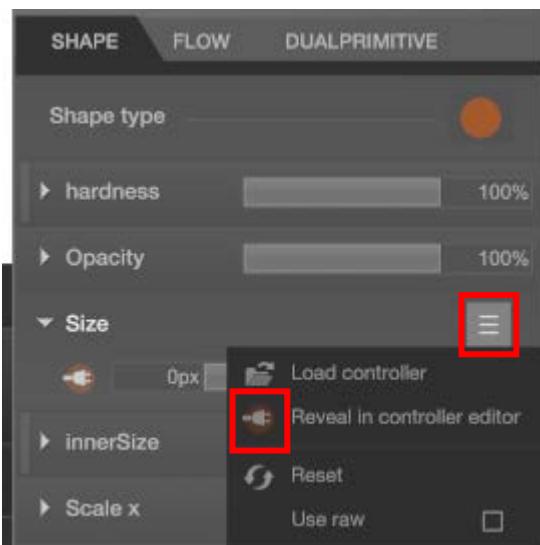
Che cosa è esattamente un controller ?

I controller sono un semplice linguaggio visivo utilizzato per personalizzare i tuoi pennelli. Vengono utilizzati per assegnare comportamenti ai parametri del pennello come, per esempio, associare alla la dimensione di un pennello alla pressione del tablet. Potete usarli per semplici ritocchi, per mettere a punto il pennello da disegno perfetto o andare oltre e creare una vasta gamma di effetti unici.

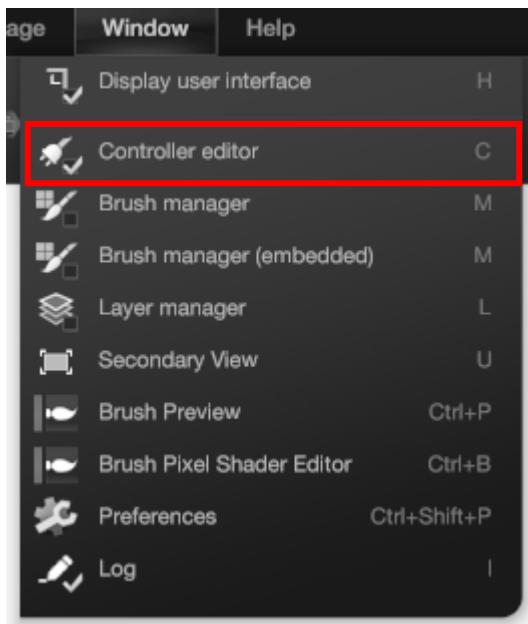
Per iniziare a vedere come funzionano, diamo un'occhiata a un pennello predefinito per vedere come è stato realizzato. Seleziona il pennello **Felt pen standard** nei pennelli preferiti di default:



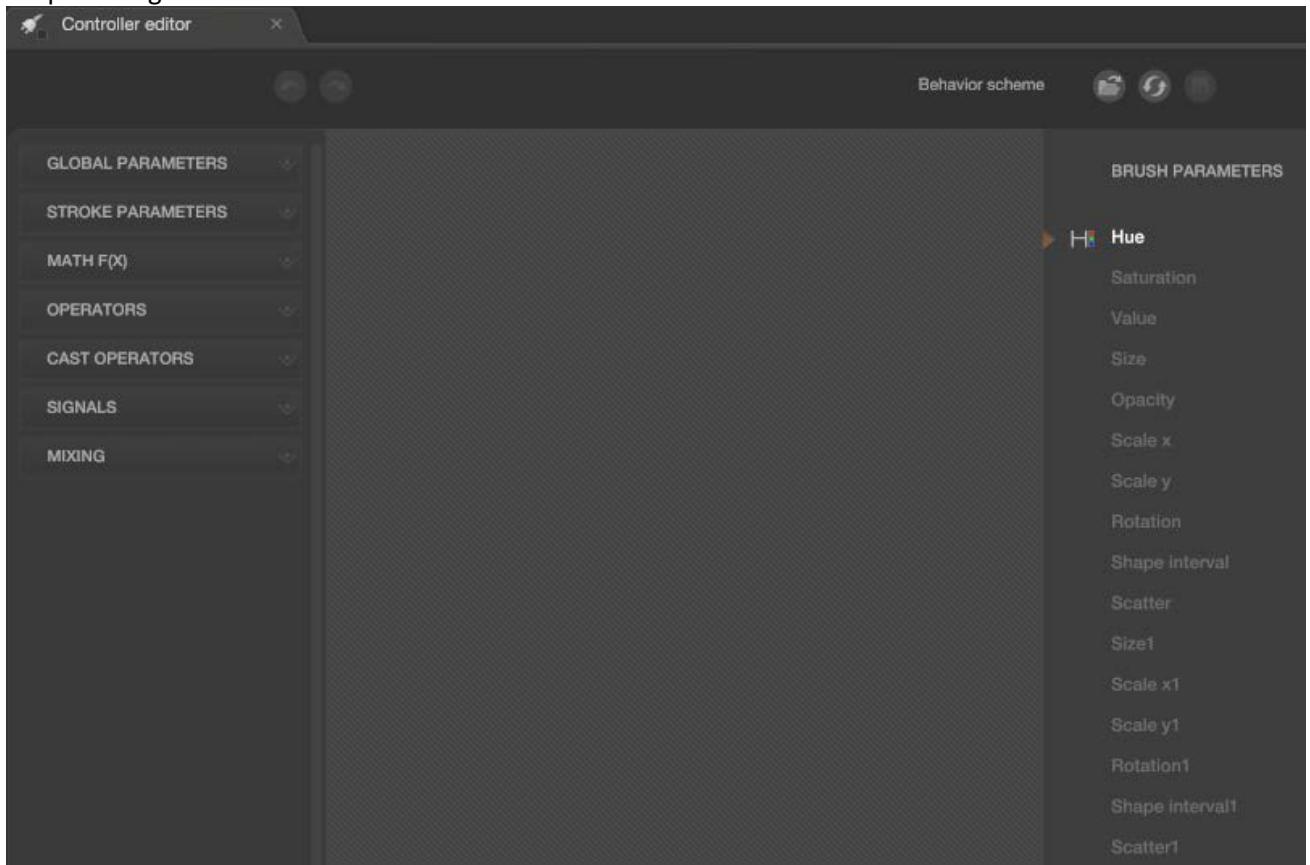
Ora bisogna aprire l'editor del controller. Ci sono diversi modi per farlo, uno di questi fare è clic sull'**icona del controller** che si trova nel menu del cursore esteso **Size** (riquadro rosso).



Puoi anche utilizzare il menu **Windows->Controller Editor**. Il tasto scorciatoia di default è "C".



Si apre la seguente finestra:



Sulla parte sinistra troviamo degli slot che contengono i nodi con cui possiamo costruire un controller.
Al centro troviamo l'area di sviluppo, il posto in cui noi visualizziamo e colleghiamo i nodi.
Sulla destra abbiamo tutti i parametri del pennello che possono essere guidati da un controller.
Cominciamo cliccando su **Size (Dimensione)** dei **BRUSH PARAMETERS** (a destra):



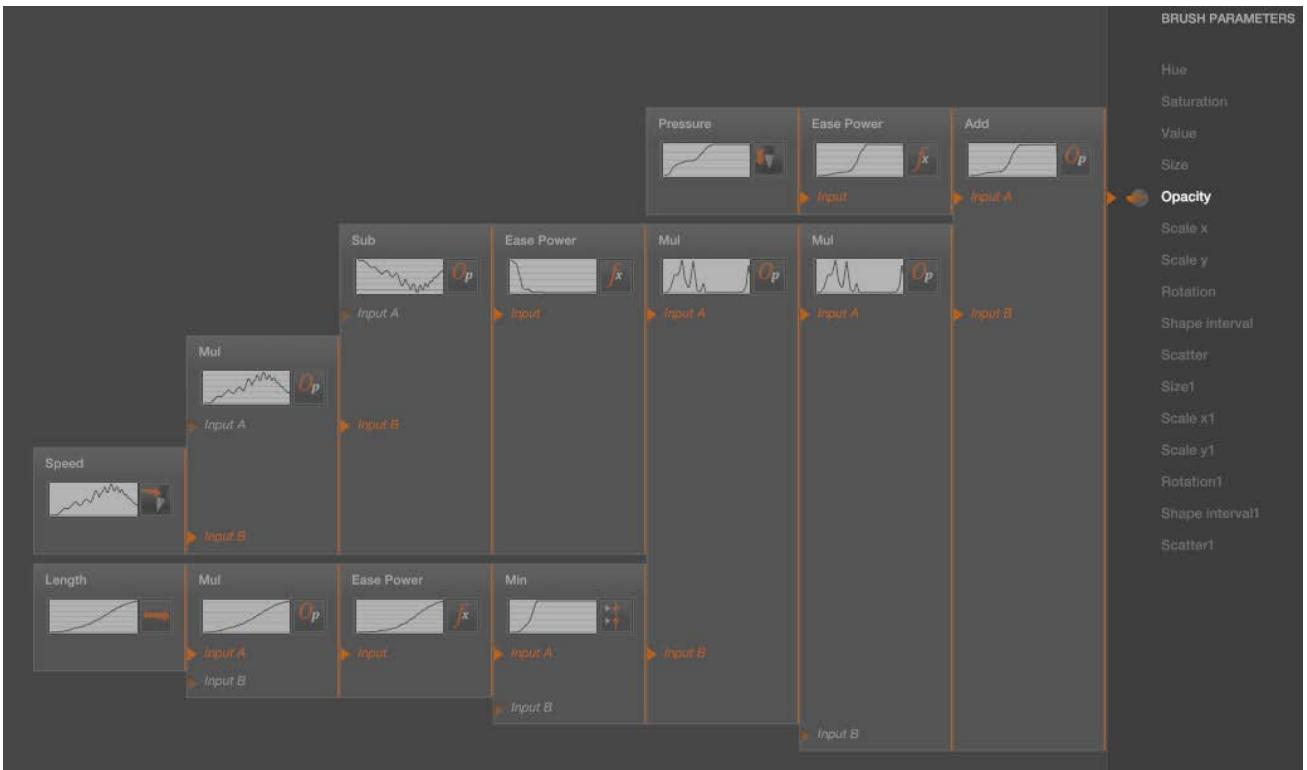
Come potete vedere esiste già un controller costituito da diversi nodi. Il primo è quello più importante: si tratta della pressione che applicate alla penna della vostra tavoletta. Gli altri due servono per rendere l'input più dinamico e fluido (parleremo di questi nodi nei tutorial successivi).

Nota:

Le versioni precedenti alla 0.357 avevano il seguente controller per Size (Dimensione):



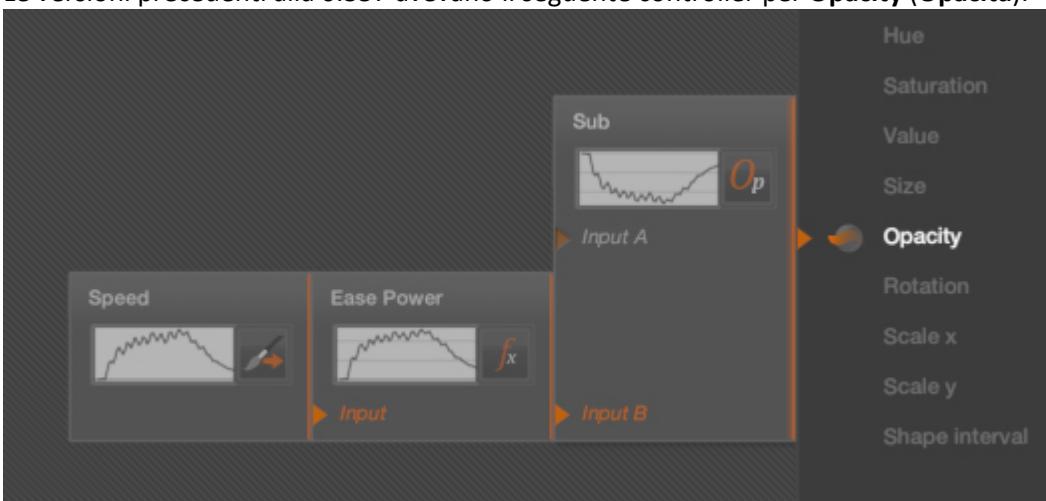
Lasciamo inalterato il controller della **Dimensione (Size)** e clicchiamo sul parametro **Opacity (Opacità)**:



In questo caso il controller è un pò più complicato: fondamentalmente questo controller inverte la velocità (**Speed**) di input della pennellata. Questo rende il pennello più trasparente quando si disegna più velocemente, come un pennarello.

Nota:

Le versioni precedenti alla 0.357 avevano il seguente controller per **Opacity (Opacità)**:

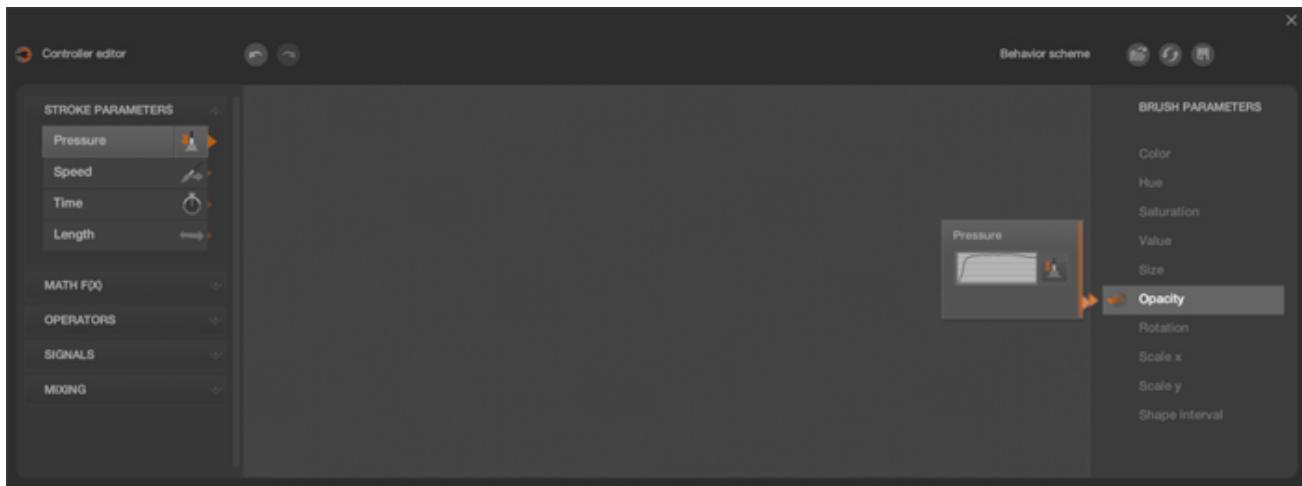


Adesso cancelliamo tutto e creiamo il nostro controller !

Clicca sull'icona a forma di Cestino che si trova nel nodo **Add** (che è il nodo posizionato più a destra). Questo eliminerà il nodo **Add** e tutti i nodi collegati ad esso che si trovano alla sua sinistra (cancellazione a cascata). Non preoccuparti: non è possibile sovrascrivere i pennelli predefiniti, quindi puoi sperimentare tranquillamente. Puoi ritornare al pennello originale selezionandolo dalla Storia dei Pennelli o riavviando BlackInk.



Ora che hai uno spazio di lavoro pulito, vai sul lato sinistro della finestra e apri lo slot "STROKE PARAMETERS". Quindi, trascina e rilascia (drag&drop) il nodo **Pressure** (**Pressione**) nell'area di lavoro.

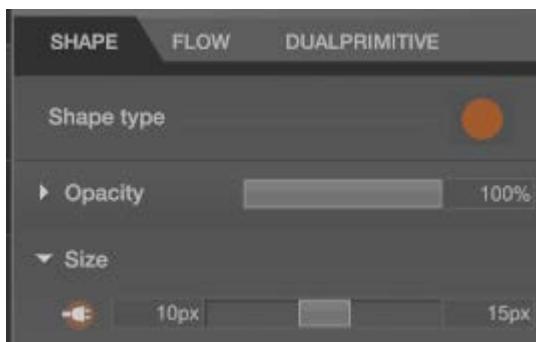


Il controller diventa attivo quando viene connesso, tramite la freccia arancione, ad uno dei parametri del pennello (**BRUSH PARAMETERS**). In controller non è attivo quando, pur trovandosi all'interno dell'area di lavoro, non è collegato ad alcun parametro del pennello.

Quando le due frecce arancioni sono vicine vengono evidenziate, allora puoi rilasciare il mouse e il nodo verrà collegato.

Questo semplice controller associa l'Opacità del pennello alla Pressione della pennellata.

Per vedere meglio il risultato modifichiamo il parametro **Size (Dimensione)** utilizzando il cursore esteso e impostando il valore minio a 10px e il valore massimo a 15px (10 quando non si preme, e 15 quando si applica più pressione allo stilo):



Proviamo a disegnare qualche pennellata:



Quindi il nostro pennello diventa più opaco quando premiamo sullo stilo!

Nota: La visualizzazione reale dei tratti del pennello avviene quando il disegno ha lo zoom 1:1 (**Ctrl Num 1**).

Questo è praticamente tutto quello che c'è da sapere per iniziare a personalizzare i pennelli: è sufficiente che un parametro del pennello (Dimensione, Opacità, ecc.), venga associato (tramite uno o più nodi) ad una variabile della pennellata (es. Pressione della penna, Velocità della penna) per ottenere variazioni del tratto il disegno. Prova con diversi parametri!

Come funziona

Come abbiamo visto nel tutorial dei primi passi, i parametri del pennello hanno un valore minimo e un valore massimo che puoi impostare nel cursore esteso.

Quello che fanno i controller è alimentare il parametro con un valore compreso tra 0 e 1.

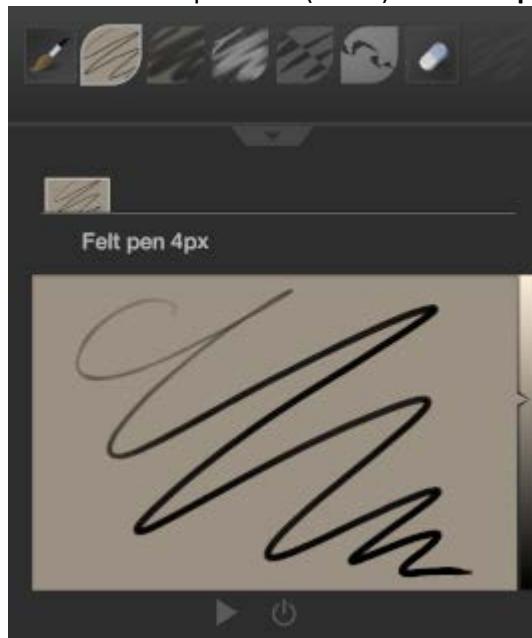
Zero (0) sarà il valore minimo del cursore e Uno (1) il massimo.

Ogni valore compreso tra 0 e 1 sarà interpolato per ottenere un valore nell'intervallo impostato.

Ora puoi sperimentare con i controller e divertirti con i pennelli!

Brush workshop 1: Pennello quadrato

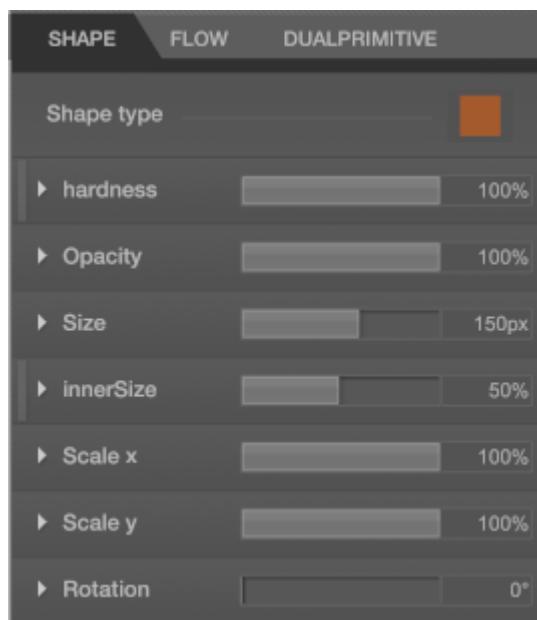
Selezioniamo il pennello (Brush) **Felt Pen 4px**:



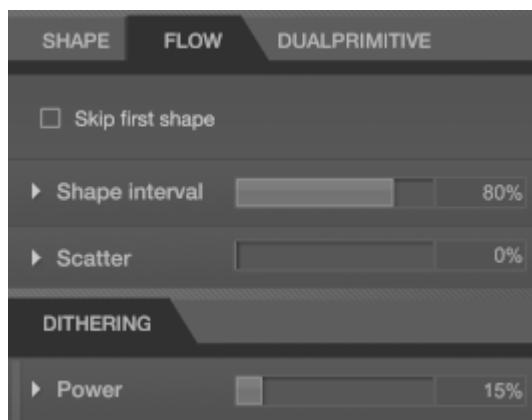
Selezioniamo il tipo di forma (Shape) quadrata per il pennello:



Selezioniamo i seguenti parametri per la forma (Shape):

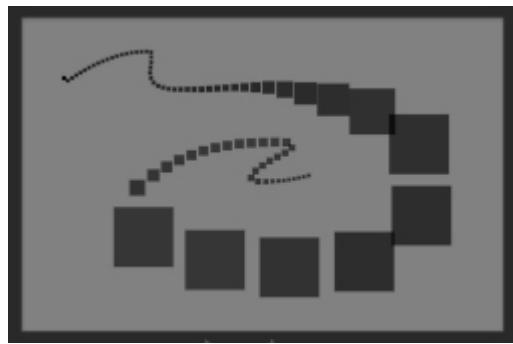


Selezioniamo i seguenti parametri per il flusso (**Flow**):

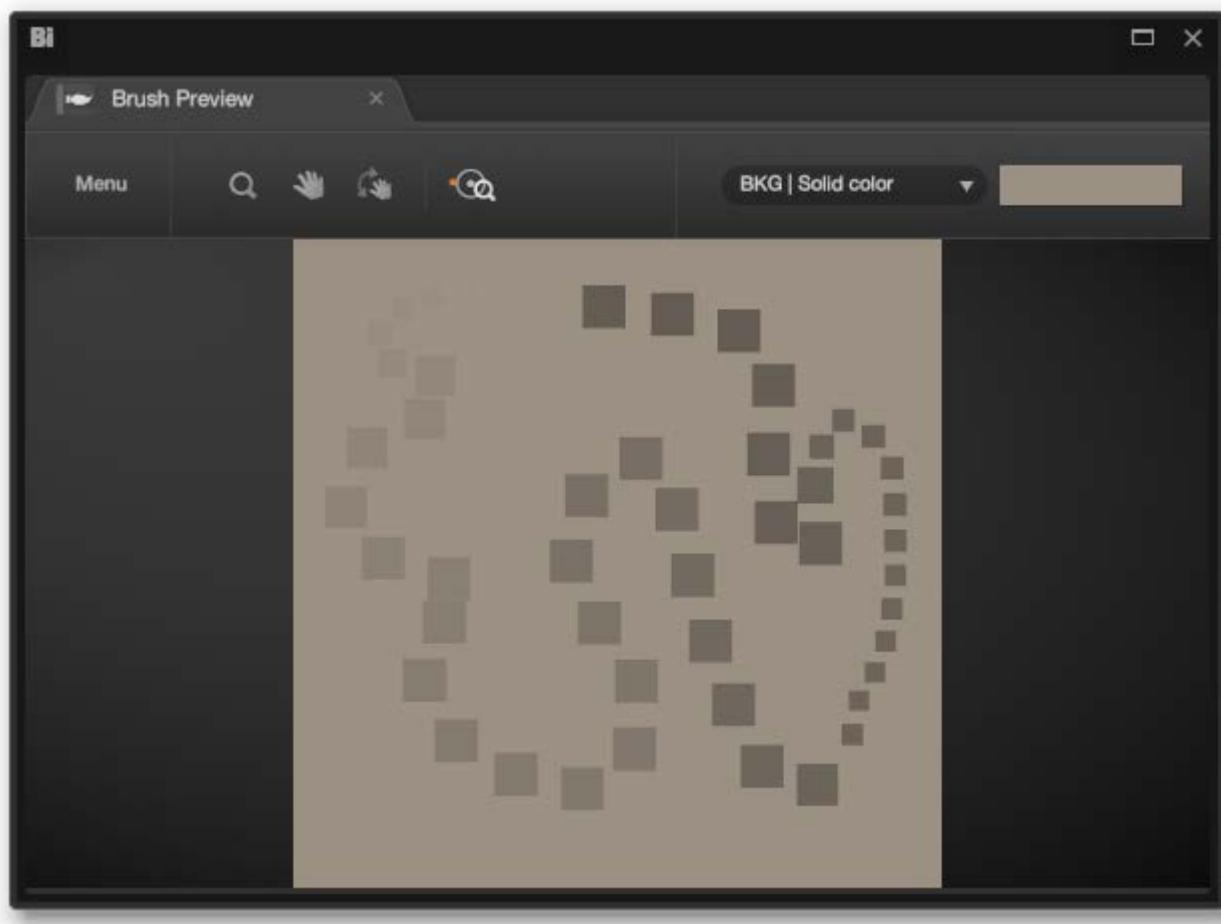


Questo parametro permette di disegnare con una sequenza di forme separate (non una pennellata con un flusso continuo).

Il pennello dovrebbe essere simile a questo:



Apri la finestra Anteprima Pennello (Brush Preview) dal menu Window->Brush Preview (Ctrl+P) e traccia una pennellata:

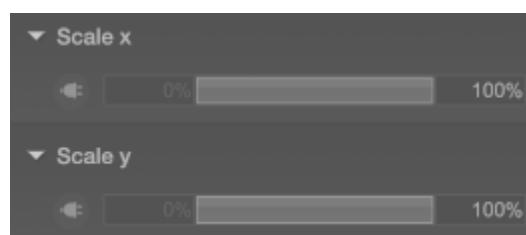


Adesso modifica i valori di minimo e massimo per la dimensione (**Size**) del pennello:



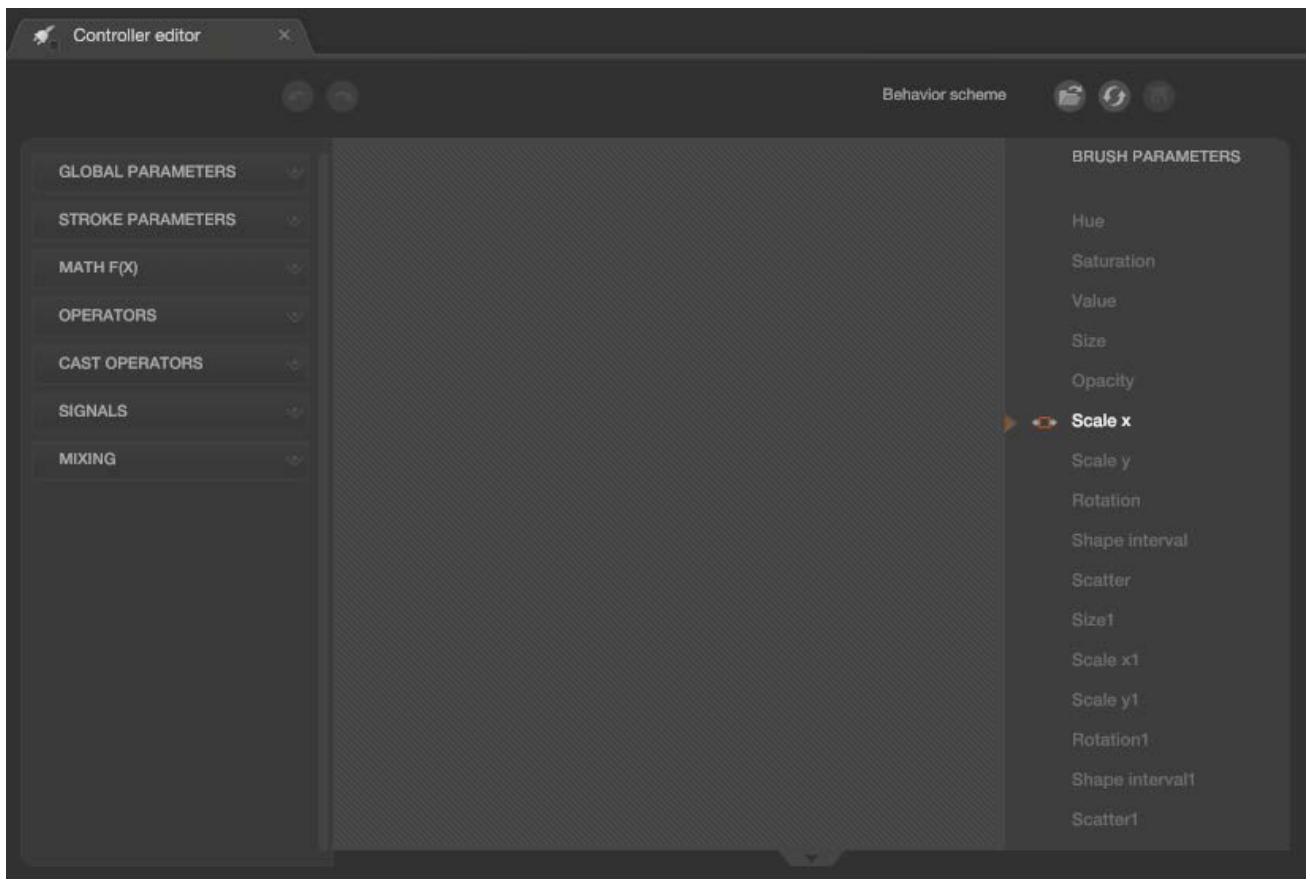
NOTA: Ogni cambiamento dei parametri del pennello modifica in tempo reale la finestra Brush Preview.

Adesso consideriamo i parametri **Size X** e **Size Y**:

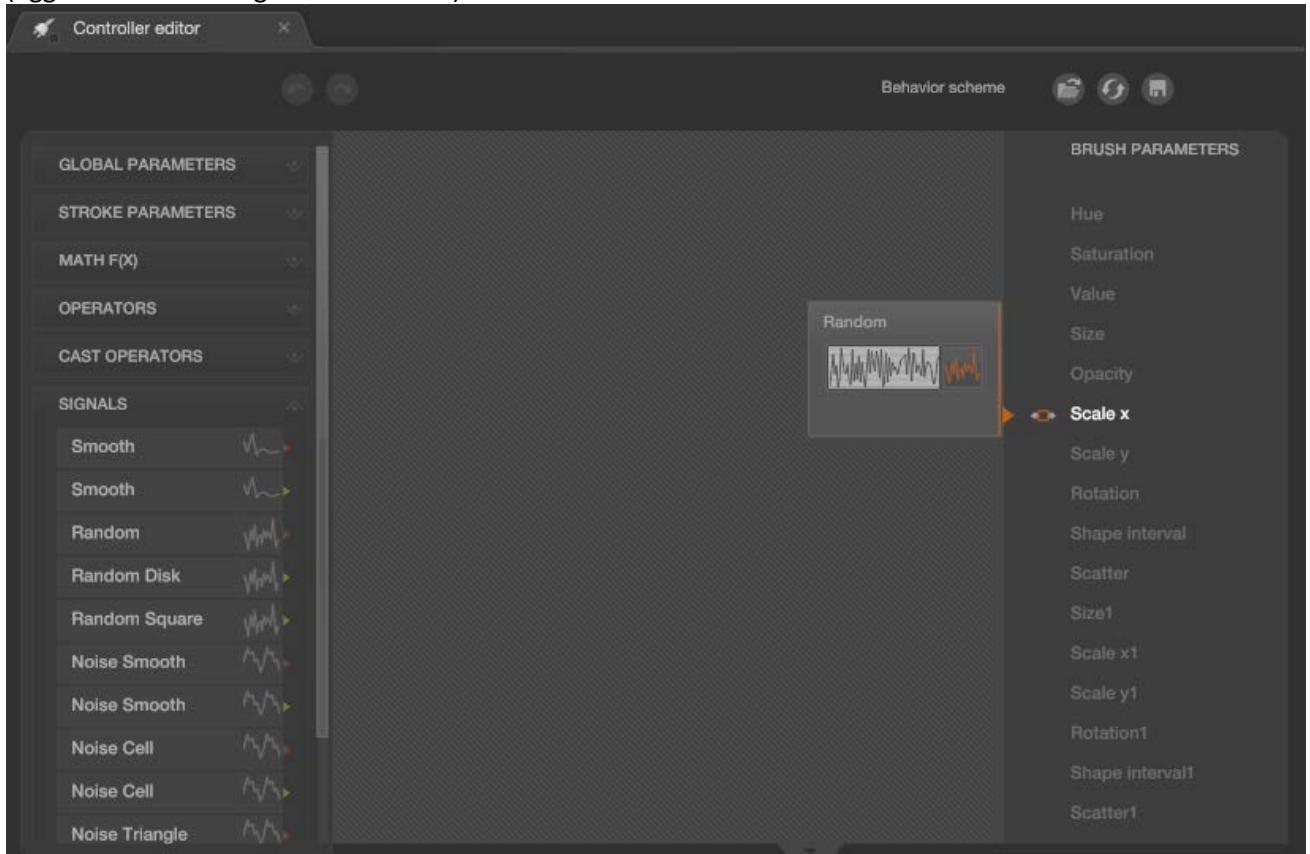


Questi vanno da 0% al 100%, ma non hanno alcun controllo (controller) che li possa gestire e modificare.

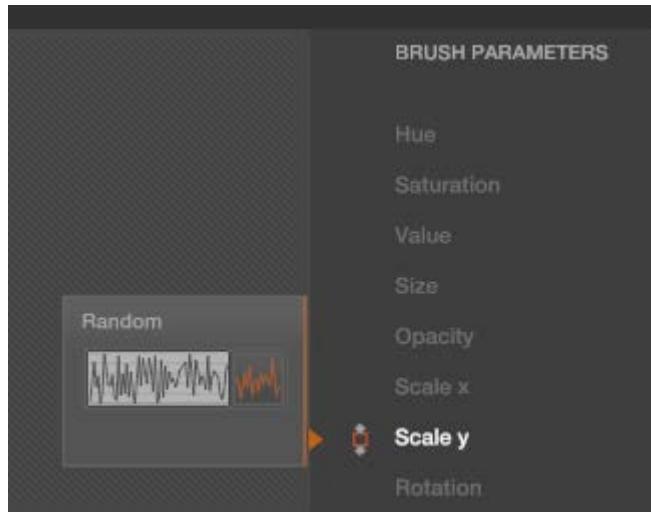
È possibile caricare un controllo esistente cliccando sul simbolo del controller oppure possiamo aprire la finestra dei controlli (**Controller Editor**) premendo il tasto “C”:



Prendiamo il blocco **Signal->Random** e lo attacchiamo (drag and drop) al parametro del pennello **Scale X** (aggancio con il triangolino arancione):

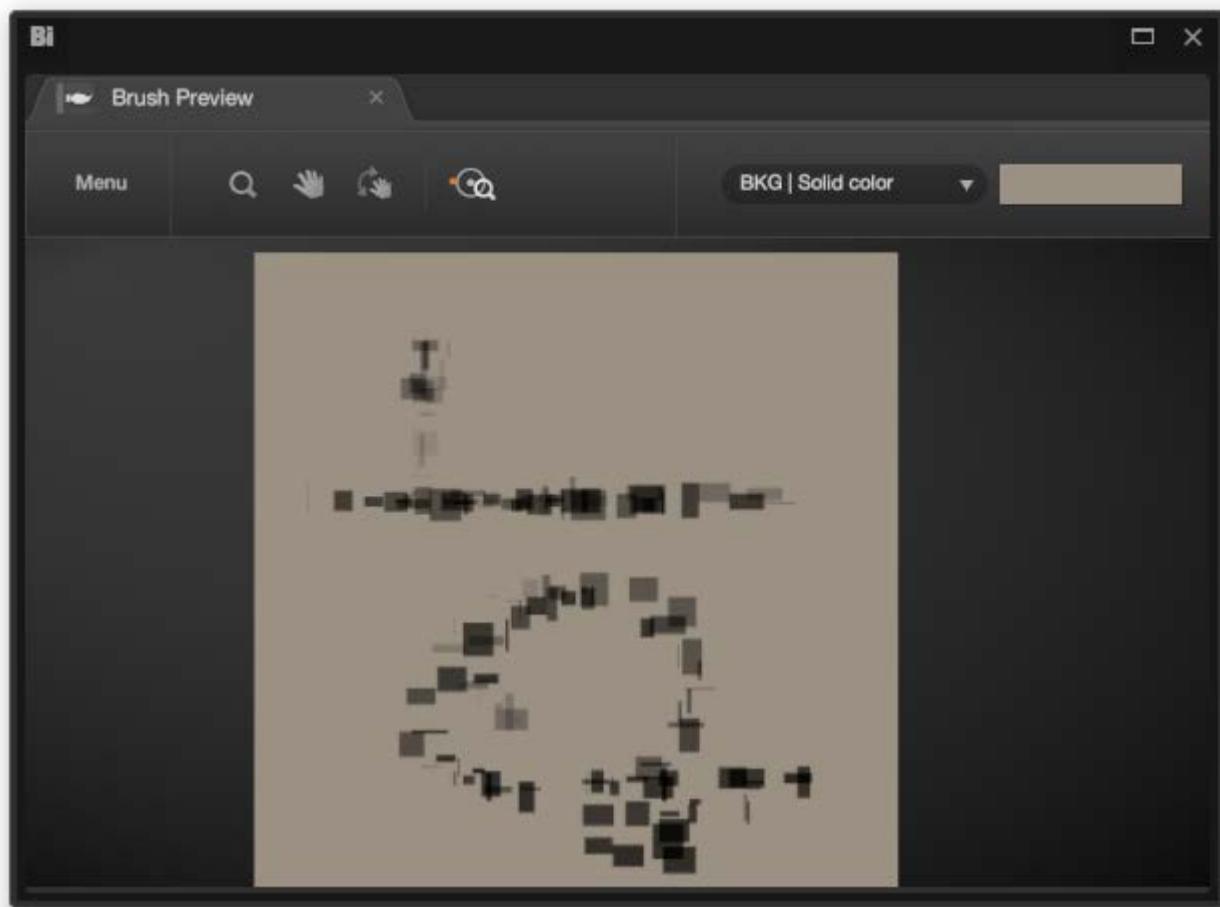


Poi facciamo lo stesso con il parametro Scale Y:



Disegniamo qualcosa sulla finestra **Anteprima Pennello (Brush Preview)**:

NOTA: possiamo cancellare il contenuto di questa finestra dal suo **Menu->Clear**.



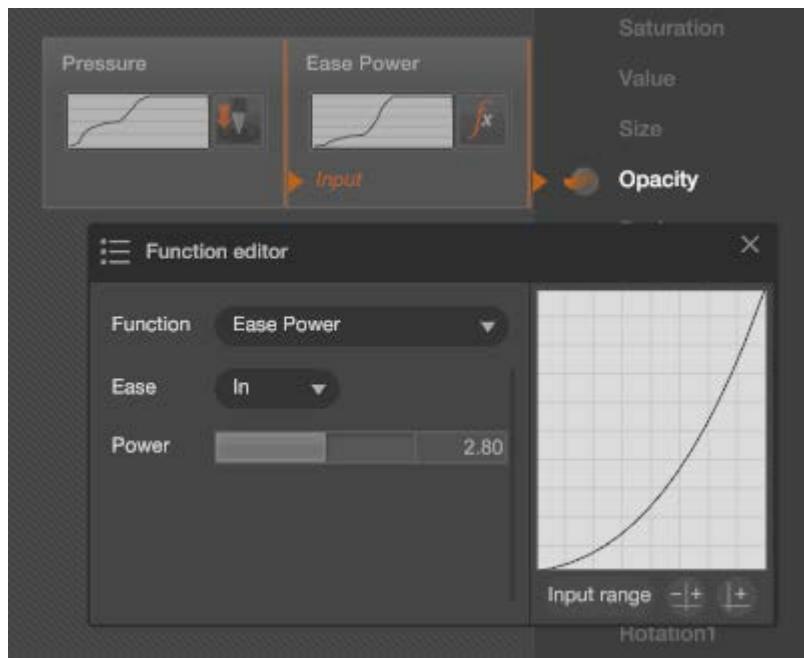
Adesso vogliamo cambiare il comportamento dell'Opacità della pennellata.

Selezioniamo il parametro **Opacity** dal **Controller Editor**, cancelliamo il controller esistente e inseriamo il nodo **Pressure** sul parametro **Opacity**.

Adesso disegnando qualcosa notiamo che è molto difficile controllare l'opacità con la pressione della penna: questo è normale, quindi dobbiamo modificare il segnale che regola l'opacità.

Prendiamo il nodo **Math->Easy Power** e colleghiamolo al parametro **Opacity** e poi modifichiamo la funzione

Easy Power (cliccando il simbolo ) nel modo seguente :



In questo modo l'opacità della pennellata è molto più graduale.

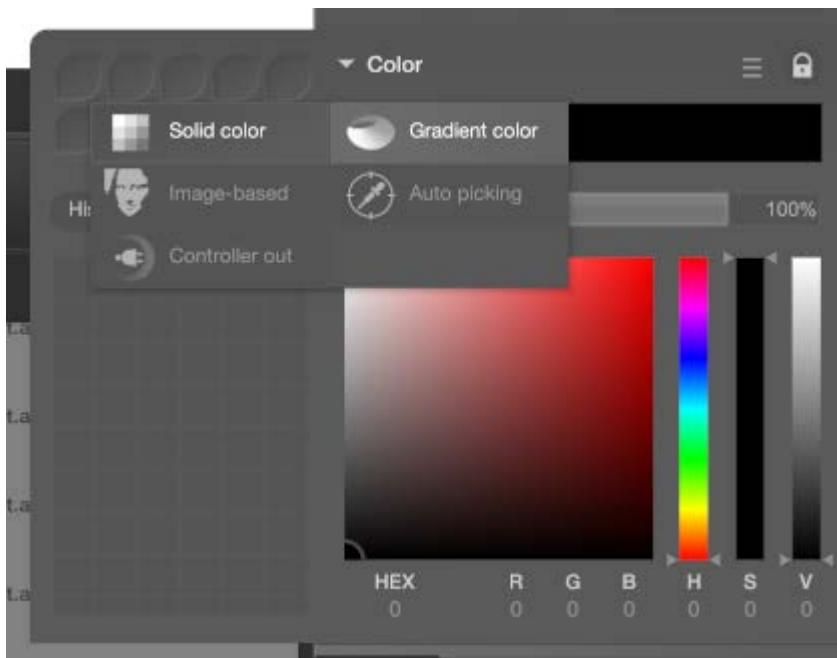
NOTA:

Questo nodo è una funzione di potenza, moltiplica un valore per se stesso un certo numero di volte. L'effetto sulla curva è questo: i valori più bassi sono ancora più bassi mentre i valori più alti salgono ancora a 1.

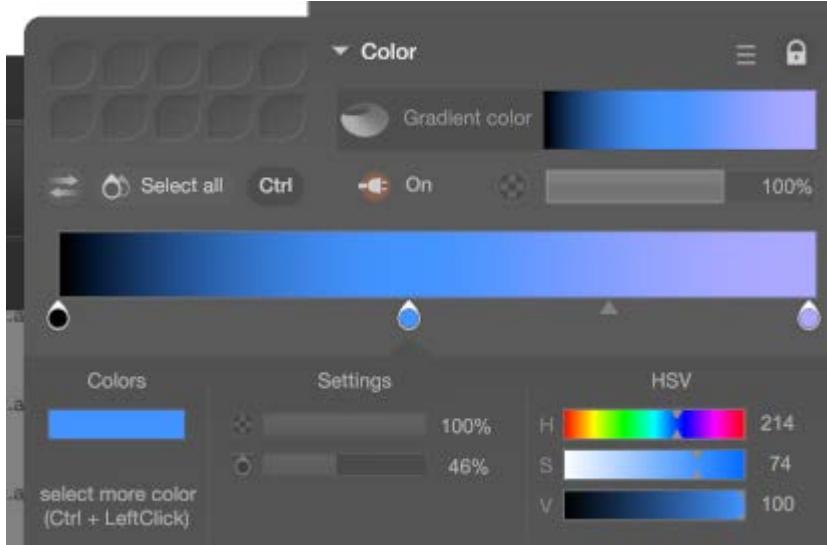
Questo è il motivo per cui a una pressione media abbiamo un'opacità più bassa, ma otteniamo ancora una piena opacità con tutta la pressione.

Adesso assegniamo il colore della pennellata in funzione della velocità di tracciamento.

Selezioniamo **Gradient Color**:



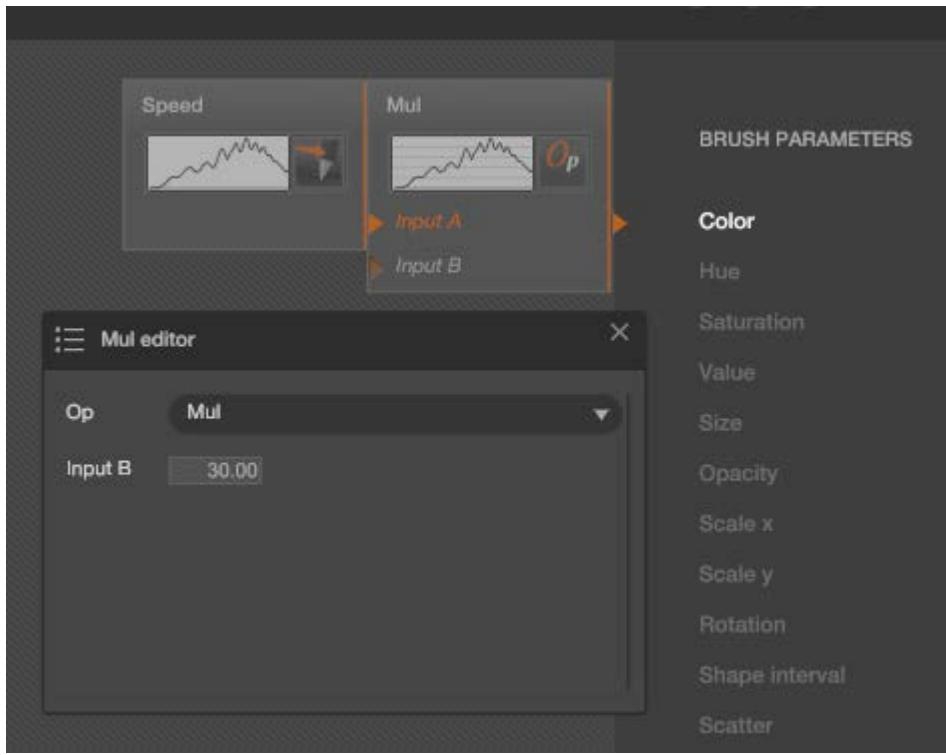
E modifichiamo i parametri:



Poi modifichiamo il controller del parametro Color passando dal controller esistente:

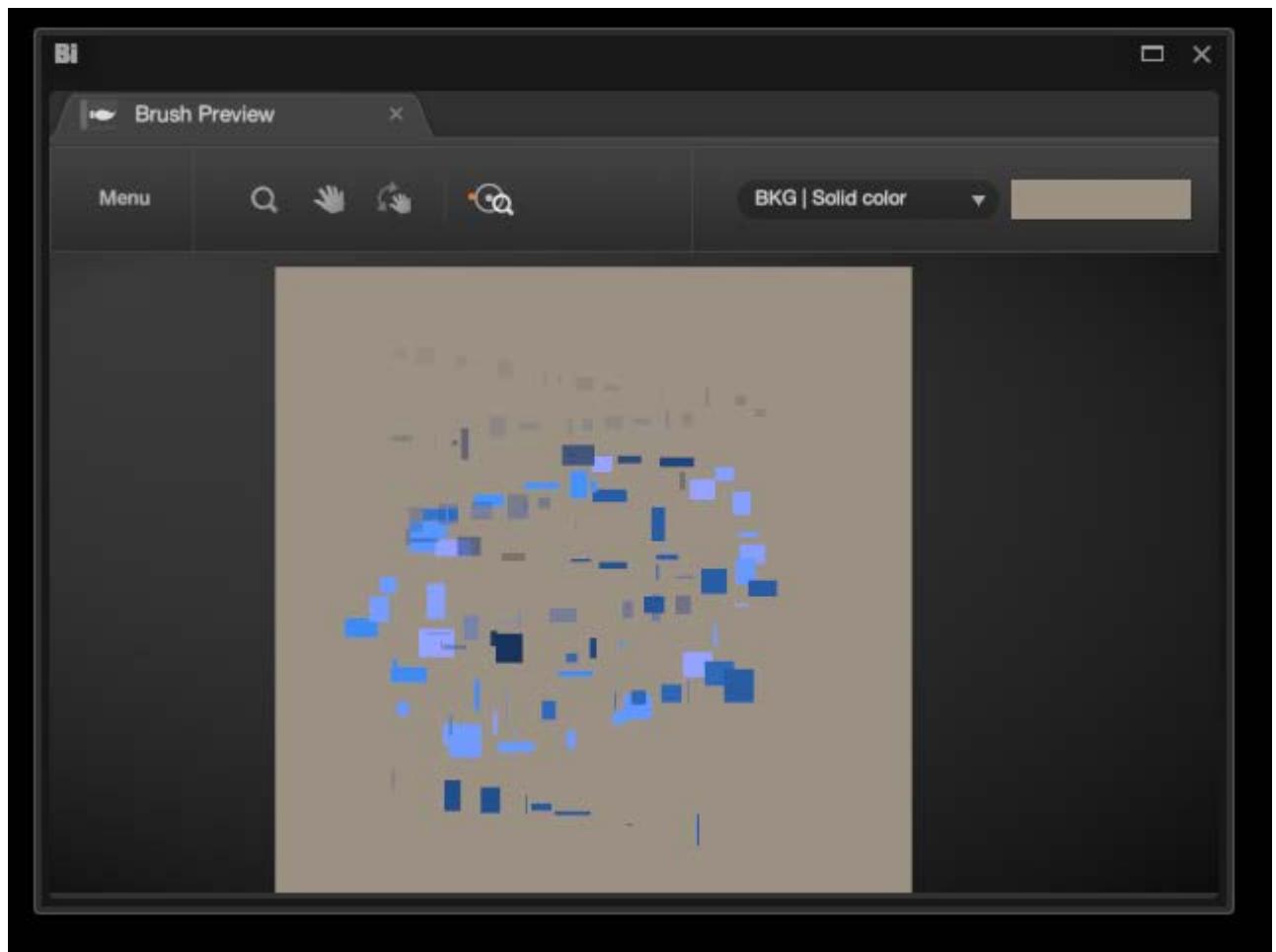


Al seguente controller:



NOTA: Se un nodo (es. Mul) ha due Input (Input A e Input B) il nodo non connesso ha in ingresso un valore costante.

Dovremmo avere il seguente pennello:



Proviamo a disegnare qualcosa:



Adesso possiamo sperimentare pennelli diversi.

Brush workshop 2: La funzione Seno

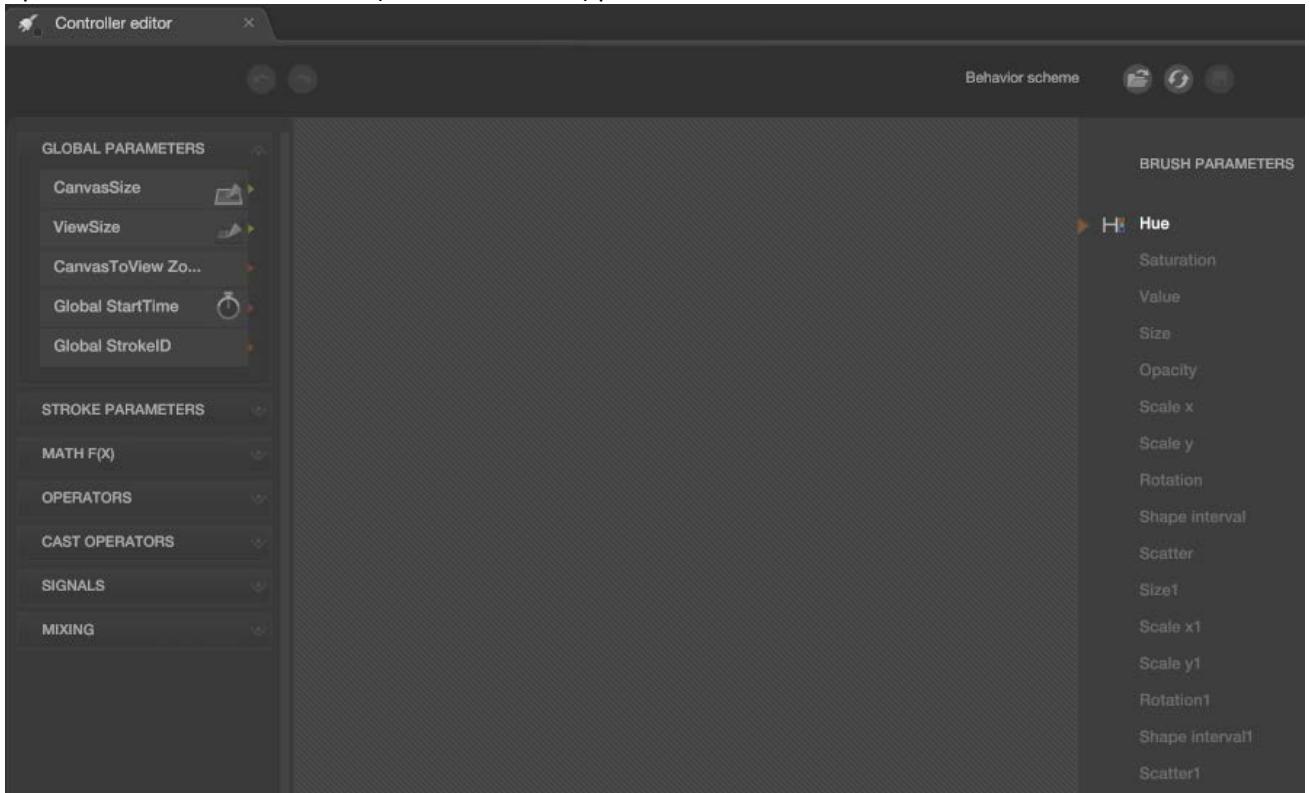
I controller sono in grado di creare pennelli che si comportano in maniera più o meno prevedibile. Stiamo parlando di Generative Art e in questo tutorial cominceremo utilizzando una semplice funzione matematica: **il seno**.

Il nostro pennello sarà semplice, ma adatto per iniziare ad allenarsi con i controller.

Cominciamo selezionando il pennello **Felt pen default**:

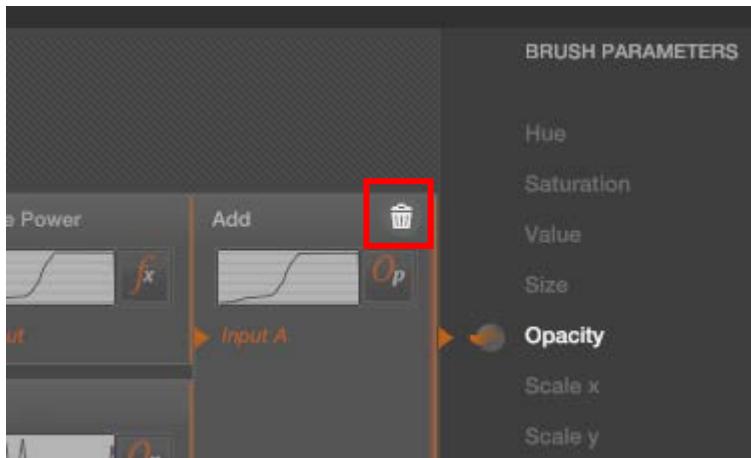


Apriamo l'editor dei controller (Controller Editor) premendo il tasto “C”:



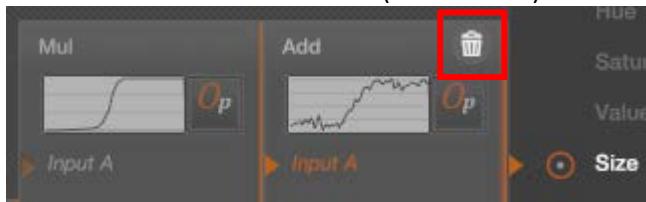
Adesso eliminiamo alcuni controller predefiniti in modo da capire bene quello che faremo successivamente.

Selezioniamo il nodo **“Opacity”** (Opacità) ed eliminiamo tutti i nodi (per fare questo basta cliccare sul pulsante “Cestino” del primo nodo a destra):



NOTA: Quando cancelliamo un nodo eliminiamo anche tutti i nodi a sinistra che sono collegati ad esso (cancellazione a cascata).

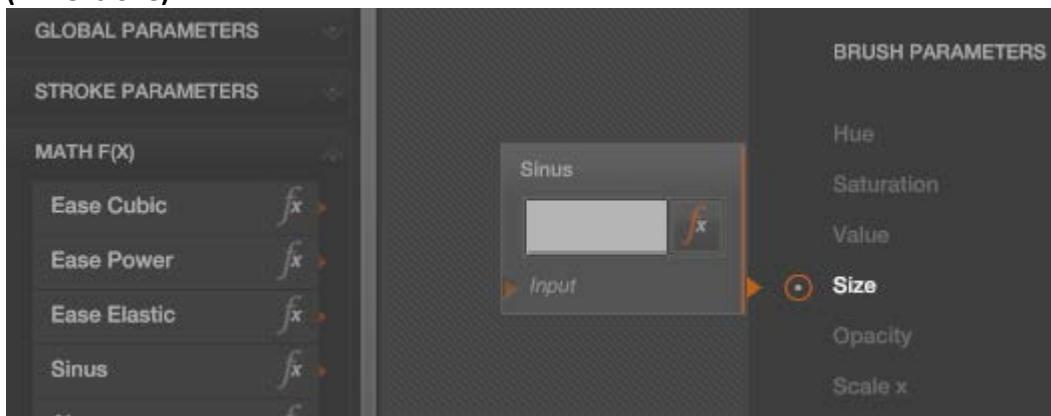
Poi selezioniamo il nodo “Size” (Dimensione) ed eliminiamo tutti i nodi:



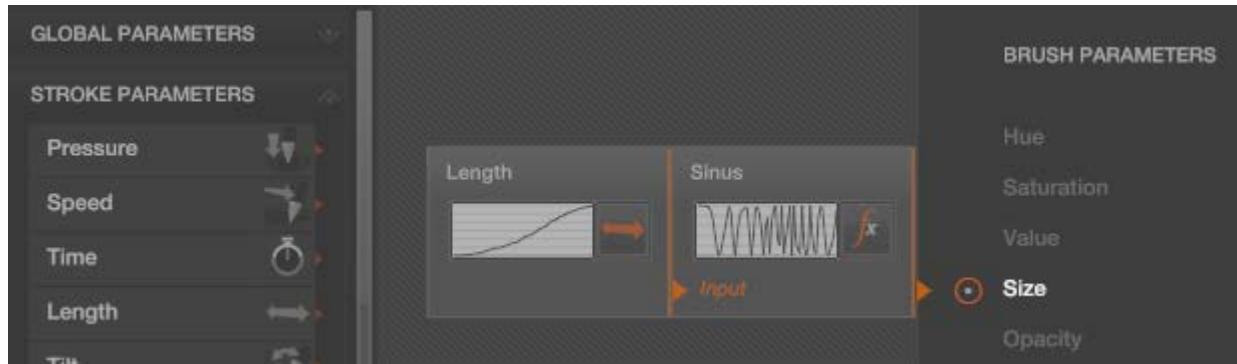
Adesso abbiamo un pennello che non cambia la Dimensione (Size) e l’Opacità (Opacity) durante la pennellata:



Adesso inseriamo un nodo, dallo slot MATH F(X) prendiamo il “Sinus” (Seno) e lo attacchiamo a **Size (Dimensione)**:



Ancora non accade nulla, perchè il Seno ha bisogno di un valore in ingresso (Input) che cresce sempre. Proviamo ad attaccare il nodo “Length” (Lunghezza) dallo slot STROKE PARAMETERS:

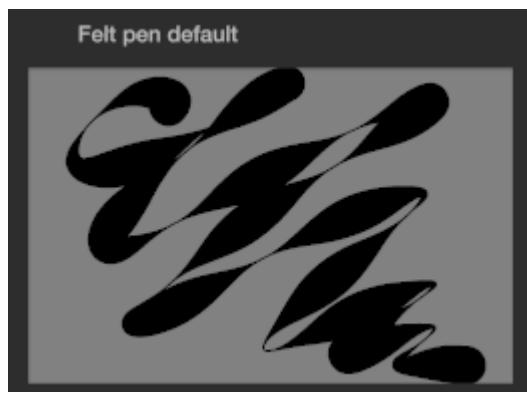


Vediamo che è cambiata l'anteprima del pennello:

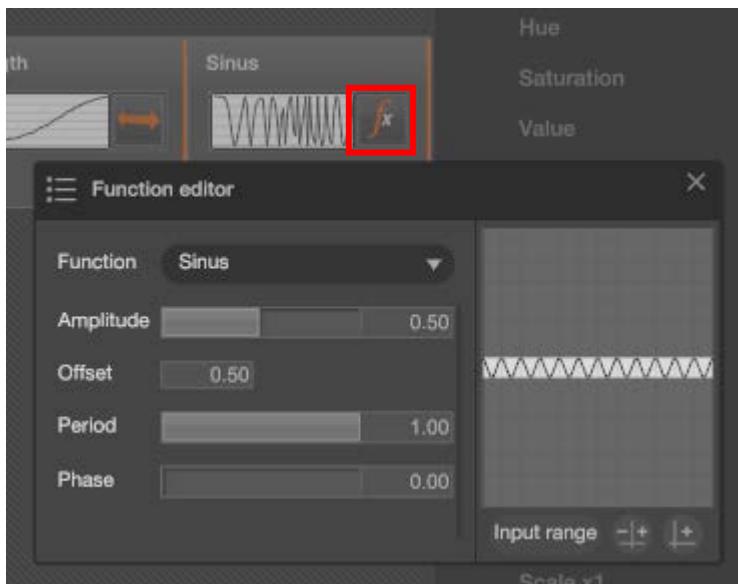


Quindi, adesso la dimensione del pennello cambia con la distanza percorsa dal pennello seguendo la funzione seno.

Modifichiamo la dimensione del pennello (Size) a 10 px e otteniamo il seguente pennello:



Adesso apriamo le opzioni del nodo Sinus cliccando sul pulsante “fx”:



La funzione Seno restituisce valori da -1 a +1, ma in questo caso restituisce valori da -0.5 a +0.5 perché abbiamo fissato la “**Amplitude**” (Ampiezza) a 0.5.

Proviamo a variare il **Periodo (Period)** e vediamo cosa accade al pennello:

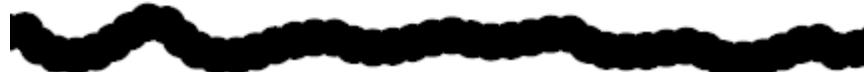
Period = 1



Period = 0.5

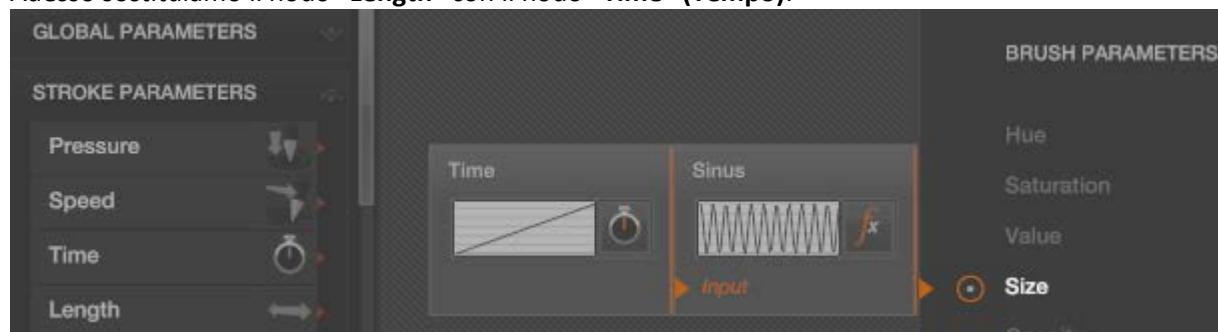


Period = 0.1



Le onde del pennello hanno sempre la stessa ampiezza perché sono calcolate utilizzando la lunghezza (“Length”) della pennellata.

Adesso sostituiamo il nodo “**Length**” con il nodo “**Time**” (Tempo):

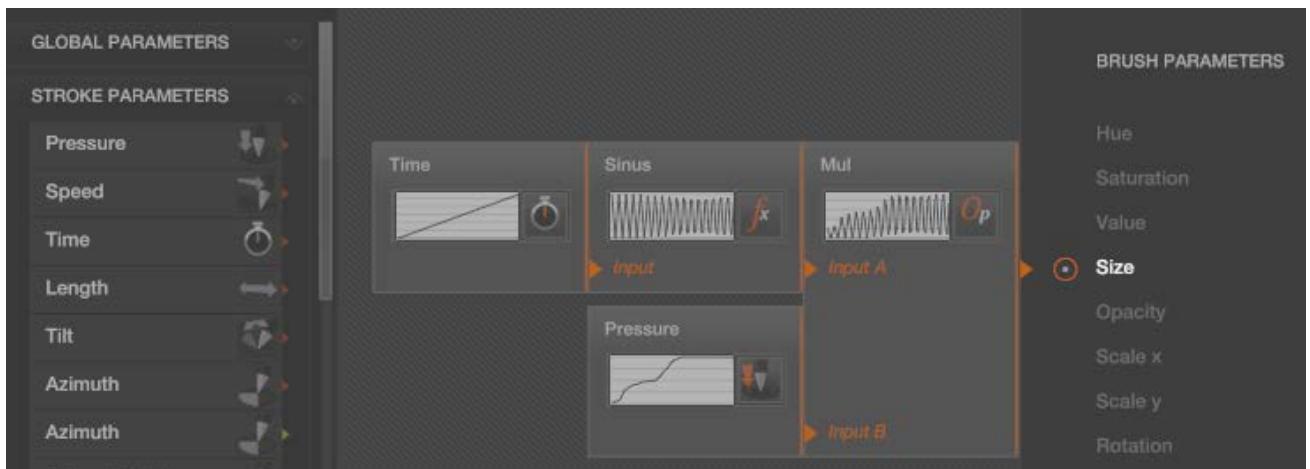


Otteniamo il seguente pennello:

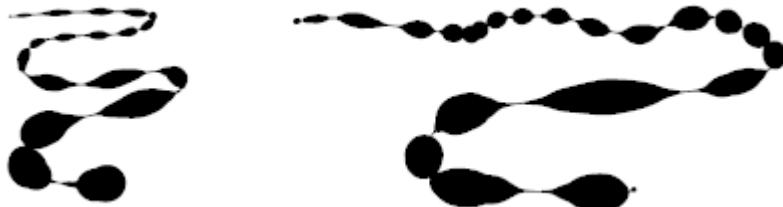


Adesso consideriamo la pressione del pennello utilizzando il nodo “**Pressure**” (**Pressione**) preso dallo slot **STROKE PARAMETERS** e il nodo “**Mul**” (**Moltiplica**) preso dallo slot **OPERATORS**.

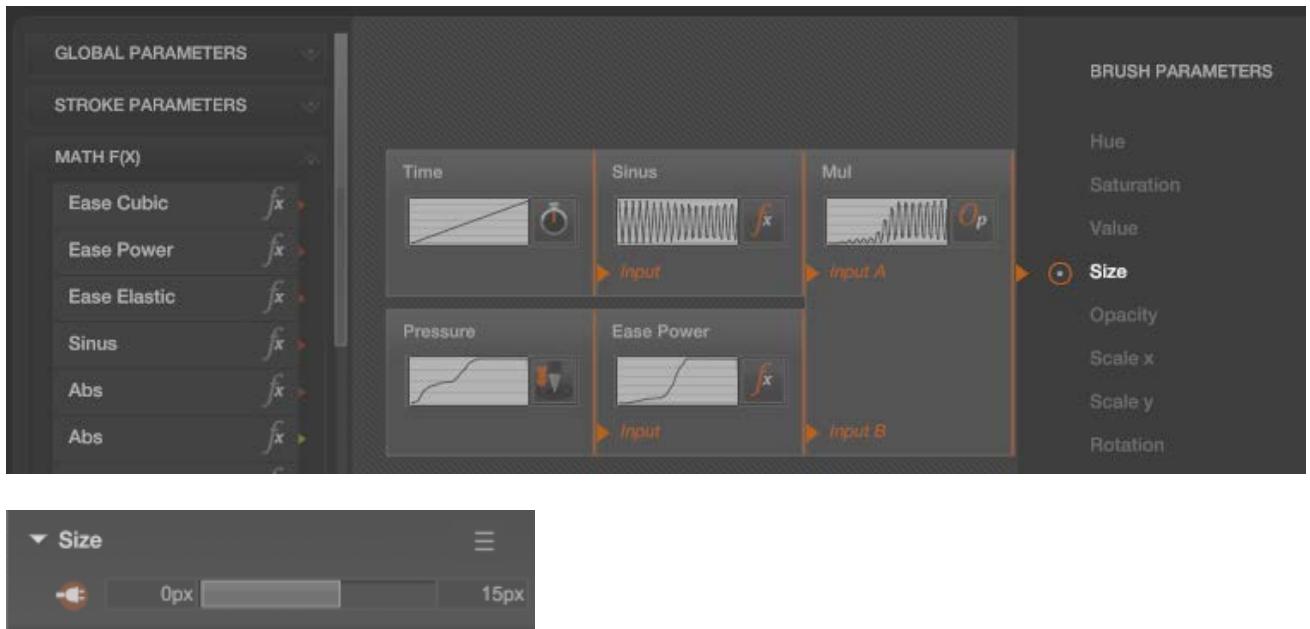
Scolleghiamo il nodo “**Sinus**” e lo attacchiamo all’ingresso A (Input A) del nodo “**Mul**”, poi attacchiamo il nodo “**Pressure**” (**Pressione**) all’ingresso B (Input B) del nodo “**Mul**” e infine ricolleghiamo tutto al parametro “**Size**” (**Dimensione**):



Questo è il tipo di pennellata che otteniamo da questo pennello (quando la pressione è bassa, allora viene ridotto il valore del Seno):



Come tocco finale aggiungiamo il nodo “**Easy Power**” (dallo slot per avere un maggior controllo dei valori bassi della pressione e impostiamo “**Size**” tra 0 1 15:



È ora di disegnare !

Questo pennello è in grado di generare gradevoli forme organiche:



Oppure simulare una penna con piuma ad inchiostro:

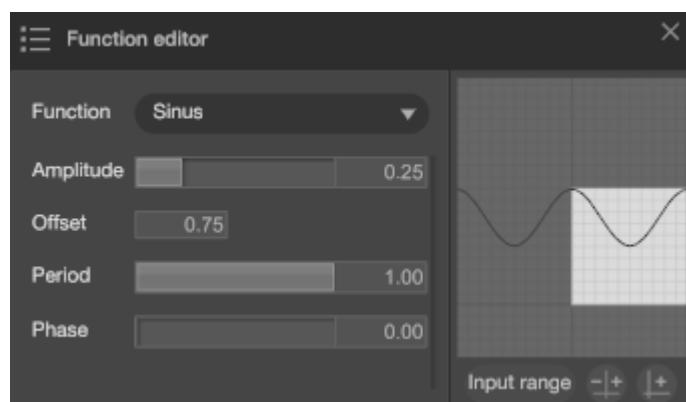


Black Ink



NOTE e SUGGERIMENTI

Questo pennello varia la dimensione da minimo a massimo seguendo i valori della funzione Seno, per ottenere un controllo maggiore possiamo cambiare i parametri del Seno con un “**Amplitude**” (**Aampiezza**) minore e un “**Offset**” (**Compensazione**) maggiore:



Questo permette di aumentare la dimensione del pennello con la Pressione anche con valori bassi per il Seno:



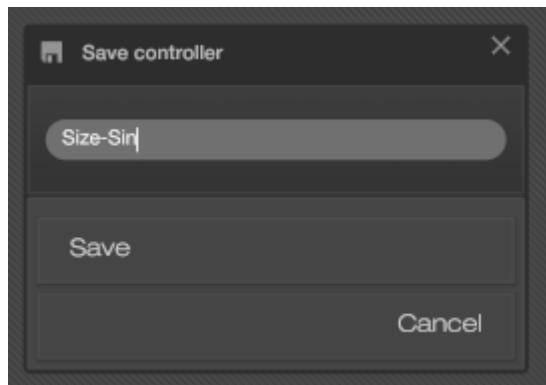
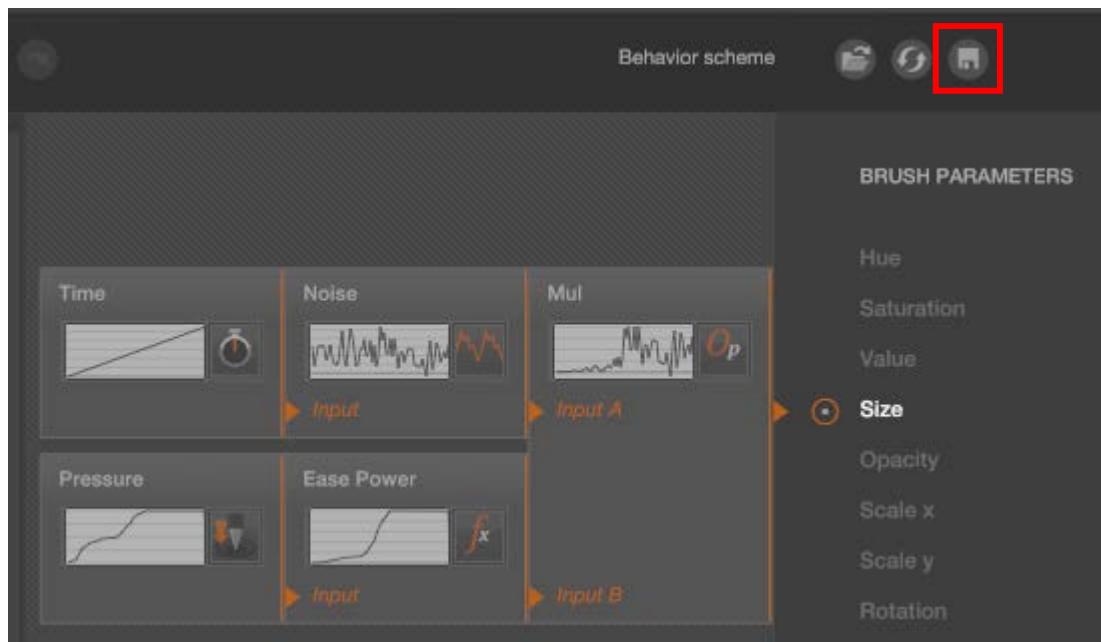
Inoltre si può provare e sostituire il nodo “**Sinus**” con un nodo “**Noise**” (Rumore) che si trova nello slot “**Signals**”. Questo darà risultati più casuali:



E, naturalmente, cercare di collegare il Seno con altri parametri e aggiungere alcuni controller alla Opacity (Opacità) per mettere a punto il pennello!

SALVATAGGIO DEI CONTROLLER

Possiamo salvare il controller appena definito premendo il pulsante evidenziato:

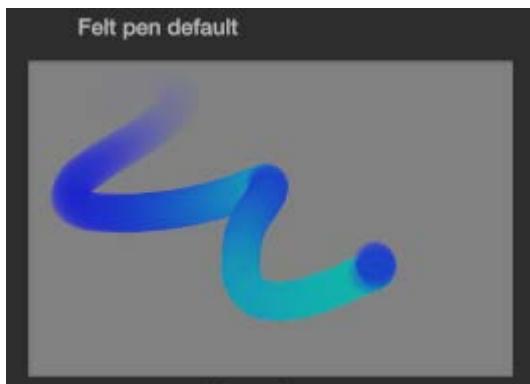


In questo modo il controller salvato può essere riutilizzato a piacimento.

Brush workshop 3: Gradienti e controller

Nei tutorial precedenti abbiamo usato i controller per gestire la dimensione e la scala del pennello, ma possiamo anche controllare il colore utilizzando i gradienti.

Selezioniamo il pennello **Felt pen default**, scegliamo **Gradient Color** (con i colori blu e azzurro), impostiamo la **Dimensione** (Size) a 10 px e infine disegniamo una pennellata direttamente sull'anteprima del pennello:



Adesso apriamo il **Controller Editor** selezioniamo l'output **Color** e verifichiamo che sia configurato nel modo seguente:



Questa è la configurazione standard per l'uso di un Gradiente come colore: viene utilizzata la Pressione come ingresso seguita da una coppia di funzioni (**Easy Power** e **Easy Cubic**) per rendere il controllo più facile.

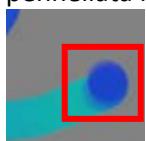
Come funziona ?

Quando la Pressione è a zero, allora viene selezionato il colore più a sinistra del Gradiente;

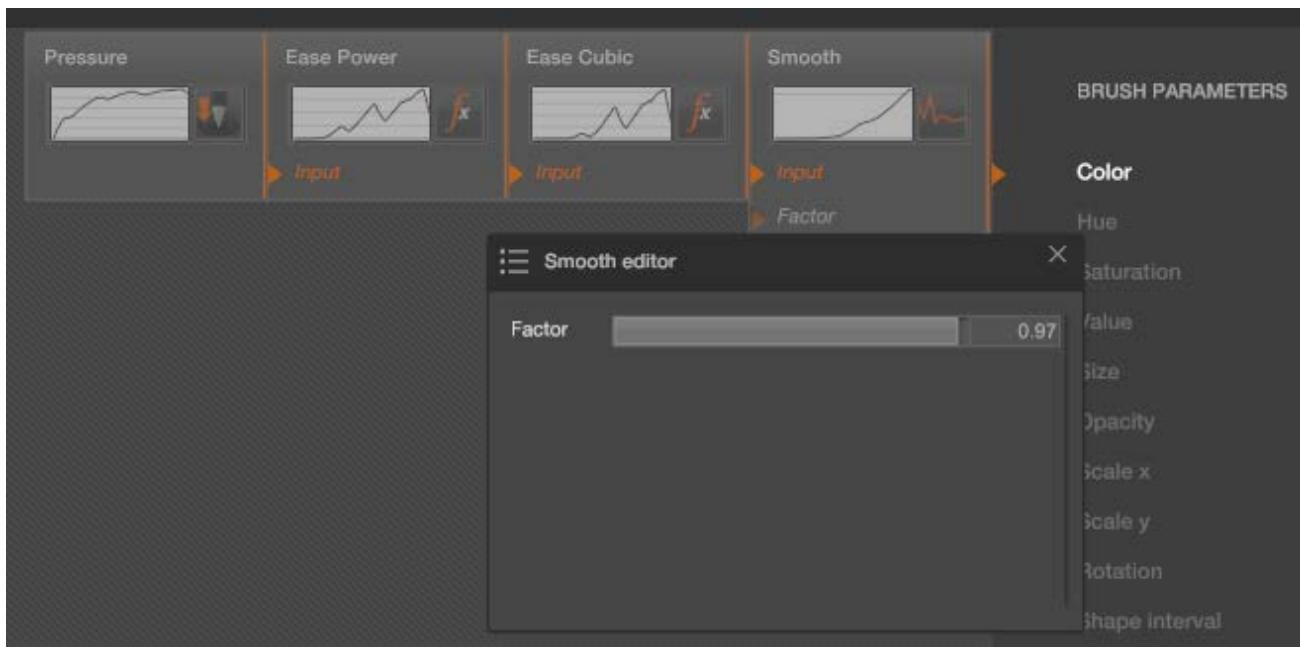
Quando la Pressione è a 1, allora viene selezionato il colore più a destra del Gradiente;

Quando la Pressione è compresa tra 0 e 1 viene interpolato il dal Gradiente il colore corrispondente.

Come si nota dall'anteprima del pennello, al termine della pennellata troviamo il colore più a sinistra del gradiente (blu) perchè in quella posizione abbiamo diminuito la pressione della penna (alla fine della pennellata la Pressione diminuisce):



Per risolvere questo problema dobbiamo modificare il controller del Colore aggiungendo un nuovo nodo: prendiamo dallo slot **SIGNALS** il nodo **"Smooth"** (Ammorbidisce) e lo inseriamo a destra alla fine della catena ponendo il suo parametro **Factor** pari a 0.97:

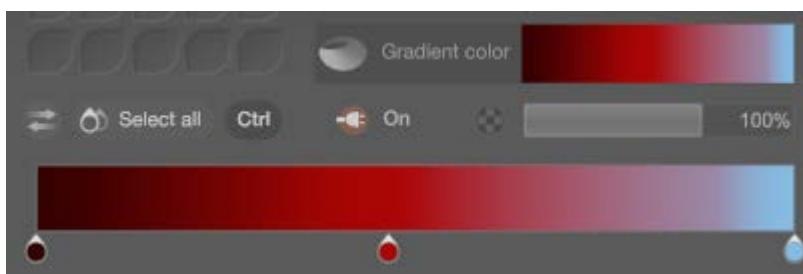


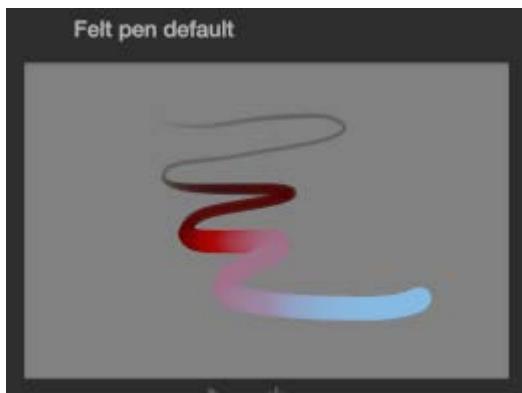
Adesso, durante la pennellata, il colore è più lineare ed abbiamo eliminato il nostro problema:



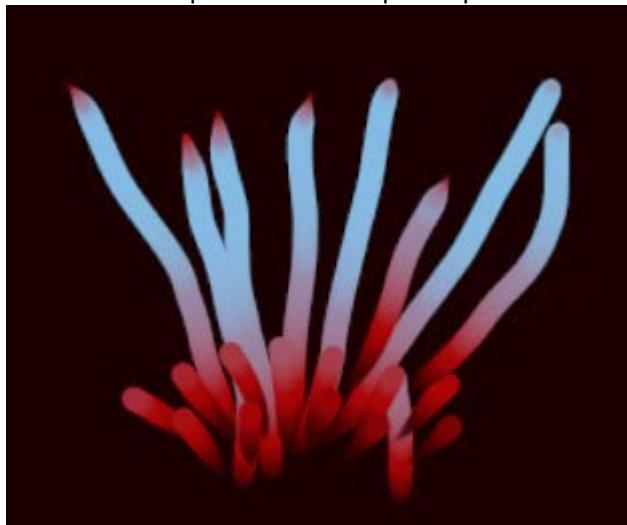
NOTA: Questo metodo (il nodo Smooth) per smussare le transizioni può essere usato anche per regolare la Dimensione (Size) nella fase finale del tratto (cioè evitando che la dimensione diminuisca rapidamente).

Adesso modifichiamo i colori del Gradiente nel modo seguente:

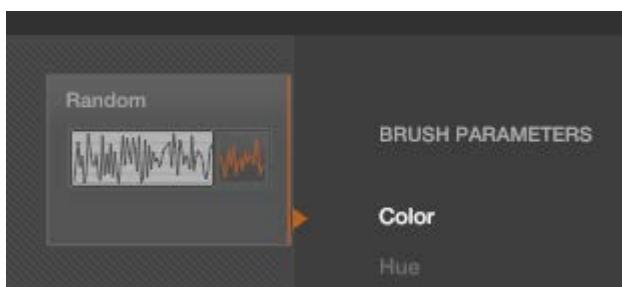




Ed ecco alcune pennellate con questo pennello:



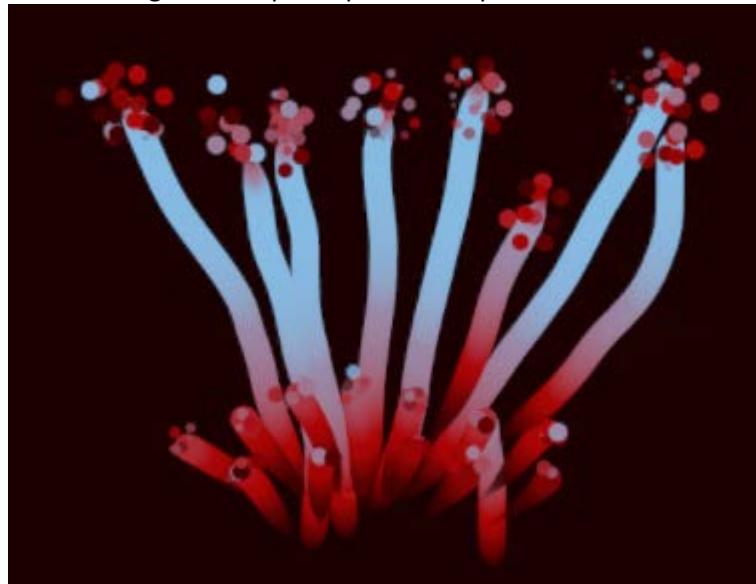
Adesso eliminiamo tutti i nodi dal parametro “Color” e inseriamo il nodo “Random” che si trova nello slot SIGNALS:



In questo modo ogni forma (Shape) del mio tratto ha un colore differente preso dal gradiente. Per migliorare il pennello apriamo il pannello “Flow” e modifichiamo il parametro “**Shape Interval**” a nostro piacimento:



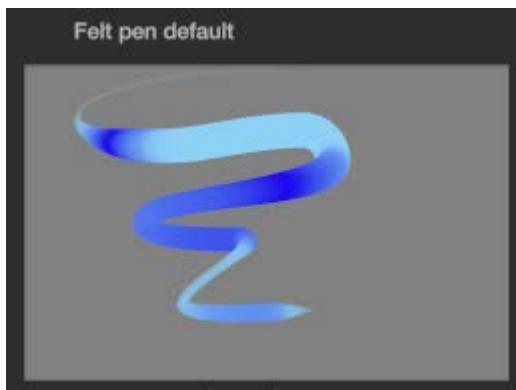
Adesso disegniamo sopra le pennellate precedenti:



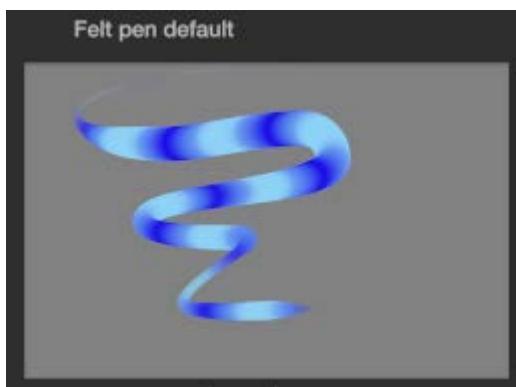
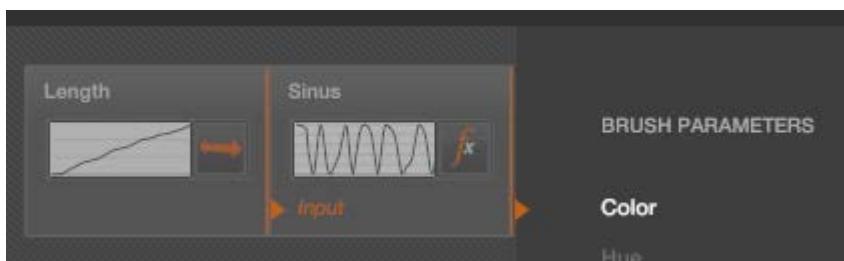
Adesso possiamo cancellare tutto e provare qualcosa di diverso (potete anche chiudere e riaprire BlackInk se l’interfaccia è diventata troppo caotica).

Selezioniamo il pennello **Felt pen default**, scegliamo **Gradient Color** (con i colori azzurro, blu e azzurro), impostiamo la **Dimensione** (Size) a 15 px e infine disegniamo una pennellata direttamente sull’anteprima del pennello:



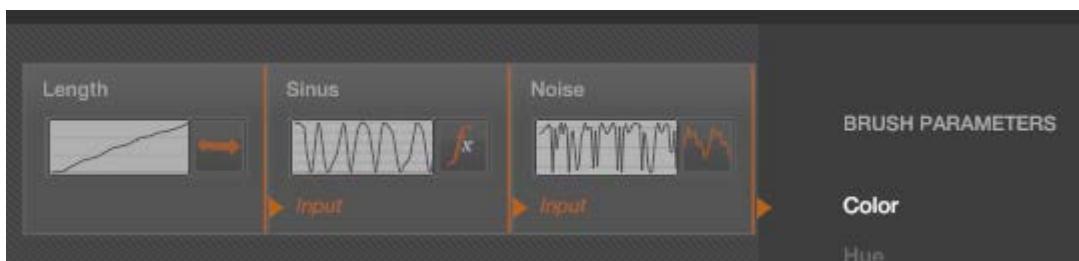


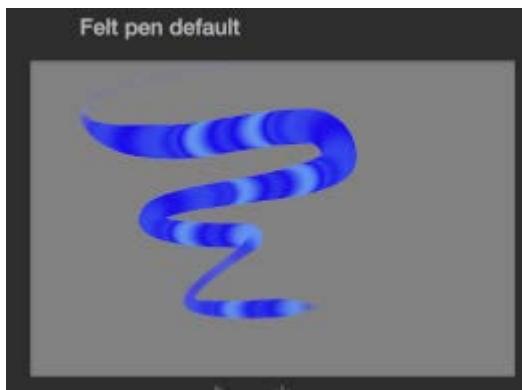
Adesso Apriamo il **Controller Editor** e attacchiamo i nodi **Sinus** (nello slot **MATH F(X)**) e **Length** (nello slot **STROKE PARAMETERS**) al parametro **Color** nel modo seguente:



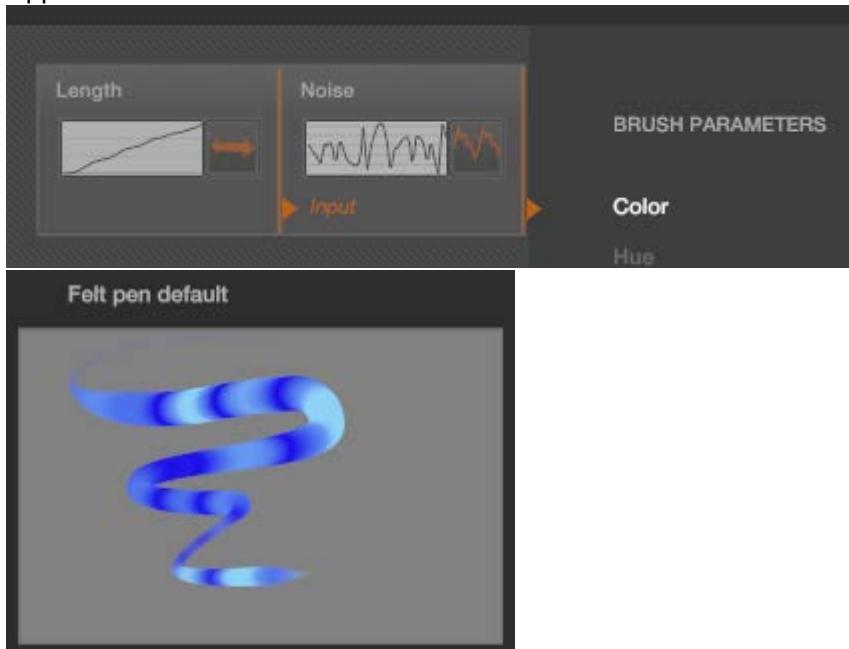
Come vedete, il colore varia continuamente da un colore estremo del gradiente all'altro (grazie alla funzione Seno).

Per ottenere un effetto meno regolare, aggiungiamo il nodo **Noise Smooth** (dallo slot **SIGNALS**); possiamo farlo in due modi diversi (ottenendo risultati diversi):



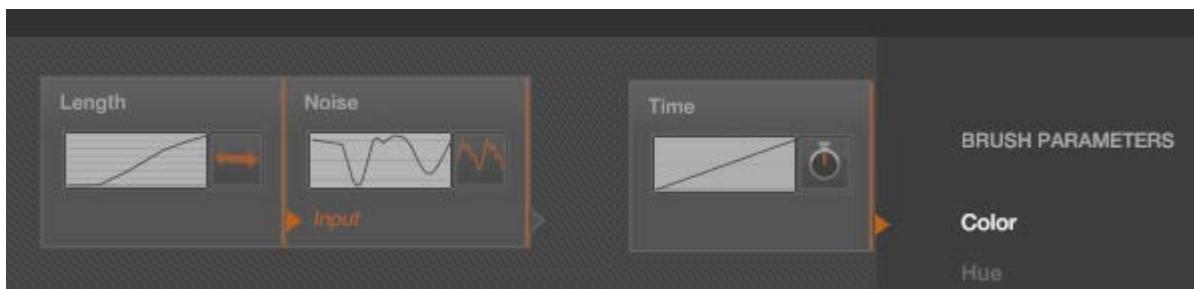


Oppure:



Proviamo ancora qualcosa.

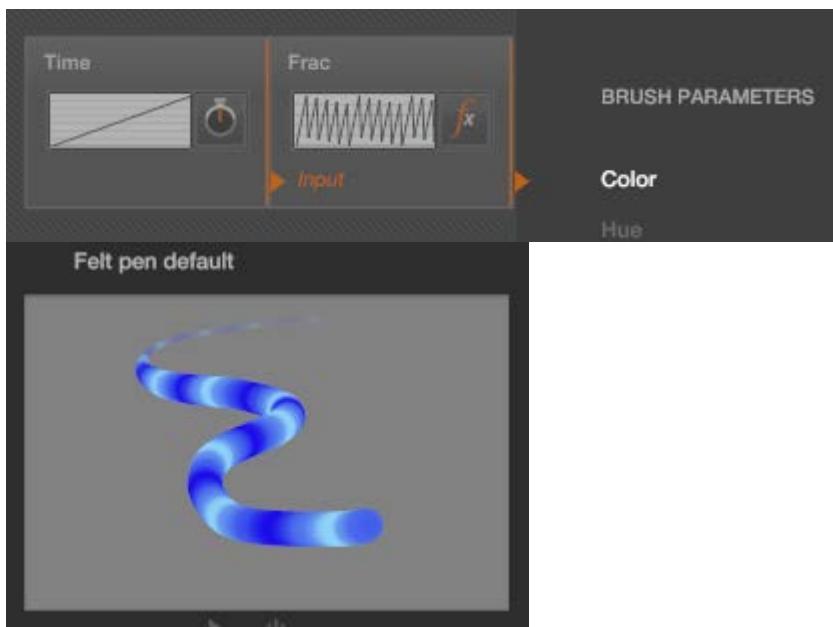
Stacchiamo il nodo **Noise** e attacchiamo il nodo **Time** (dallo slot **STROKE PARAMETERS**) al parametro **Color**:



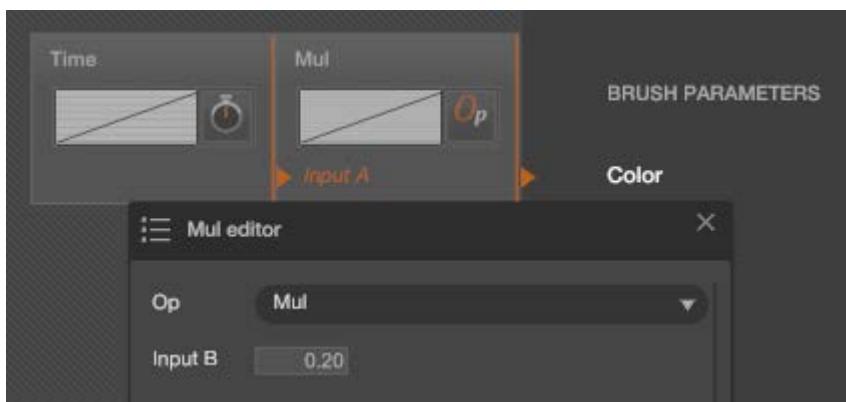
L'anteprima del pennello non sembra mostrare variazioni di colore (tranne all'inizio del tratto):



Questo è dovuto al fatto che il nodo Time raggiunge il valore 1 (uno) molto rapidamente e, poichè ogni valore maggiore di 1 produce il colore a destra del gradiente, la nostra pennellata mostra un unico colore. Per restringere il valore del nodo Time nell'intervallo [0..1] aggiungiamo il nodo **Frac** (dallo slot **MATH F(X)**). La funzione Frac prende in ingresso un numero decimale e restituisce la parte decimale, in altre parole questa funzione restituisce sempre valori compresi tra [0,1].



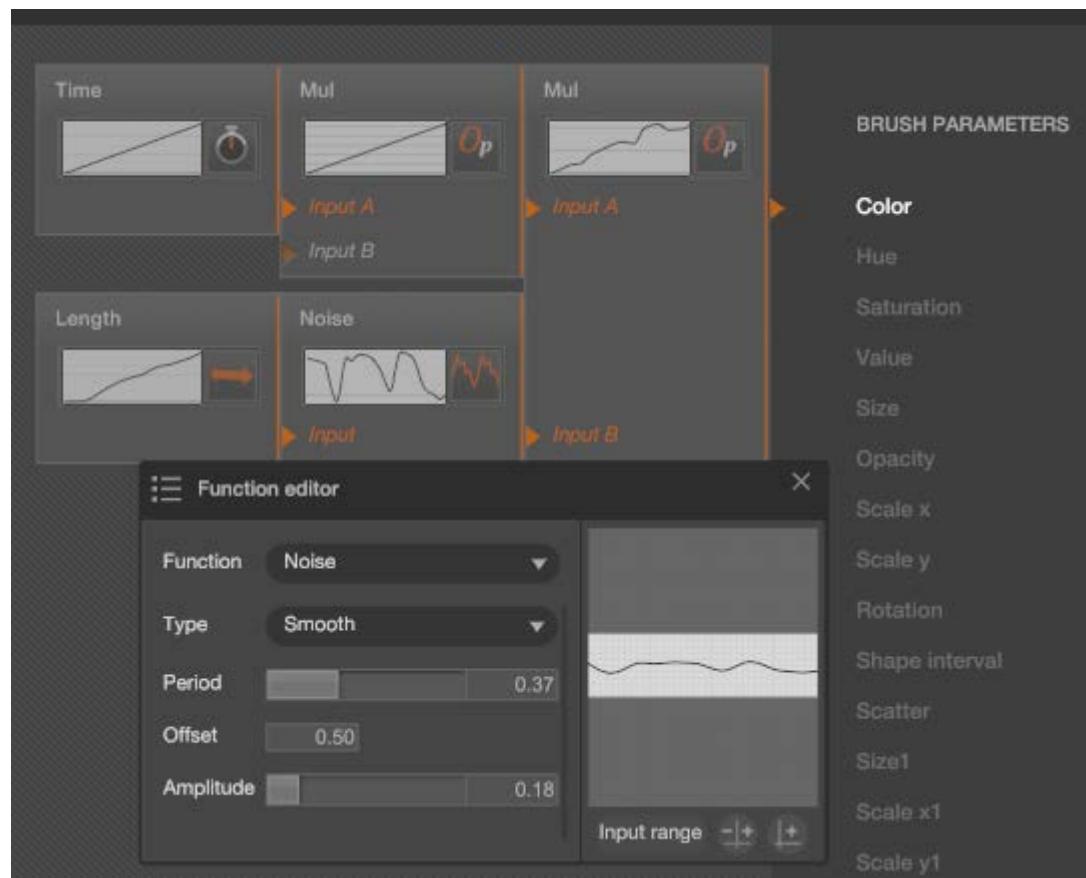
Adesso vogliamo che il gradiente sia ripetuto una sola volta durante la nostra pennellata. Per fare questo sostituiamo il nodo **Frac** con il nodo **Mul** (dallo slot **MATH F(X)**) e settiamo il suo **Input B** a 0.2:





Adesso il gradiente viene ripetuto lungo la pennellata una volta soltanto (poi il pennello assume sempre il colore a destra del gradiente).

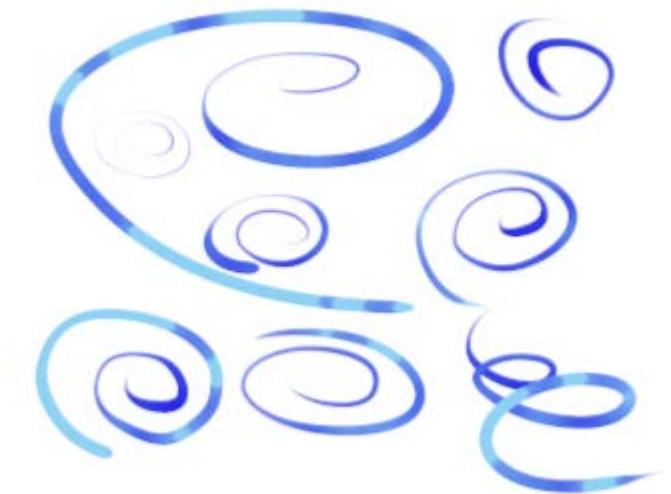
Completiamo questo controller inserendo i nodi **Length(STROKE PARAMETERS)**, **Noise (SIGNALS)** e un altro nodo **Mul (MATH F(X))**. Inoltre impostiamo il parametro **Amplitude** del nodo **Noise** al valore 0.2:





In questo caso il nostro valore cresce con il passare del tempo, ma non in linea retta poichè il valore viene modificato (in più o in meno) dalla funzione Noise. La riduzione dell' Ampiezza (Amplitude) evita i valori nulli (0), in modo che la funzione Mul non dia un risultato nullo, ma restituisca lo stesso colore.

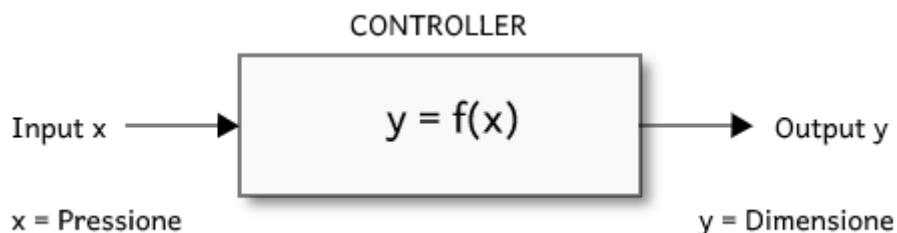
Ed ecco il risultato di alcune pennellate:



Brush workshop 4: Matematica e controller

Fondamentalmente un controller mette in relazione un parametro di ingresso della pennellata (es. Pressione) con un parametro di uscita della pennellatta (es. Dimensione).

Questa relazione è una funzione matematica e possiamo schematizzare un controller nel modo seguente:



In pratica il controller rappresenta la funzione $f(x)$, quindi i nodi del controller rappresentano le operazioni matematiche che possiamo utilizzare per modificare il segnale di ingresso (Input) per ottenere l'uscita (Output) desiderata.

ESEMPIO

Supponiamo di voler associare la Pressione della penna con la Dimensione del tratto in maniera inversamente proporzionale (una bassa Pressione implica una Dimensione maggiore e viceversa): in pratica dobbiamo invertire il range del segnale di ingresso.

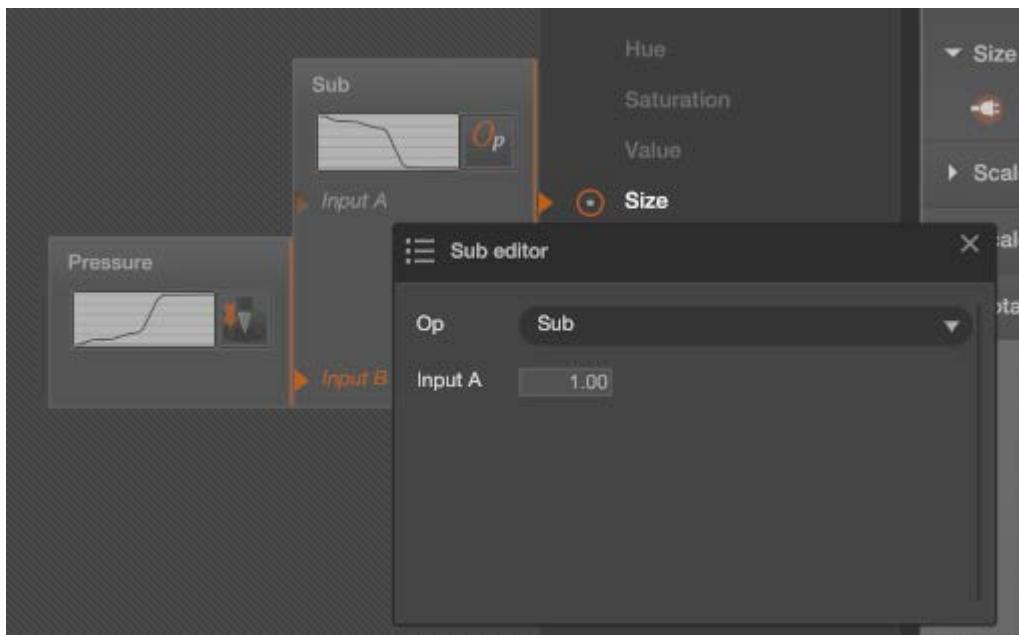
I controller hanno un segnale in ingresso che varia nell'intervallo $[0..1]$.

Se invertiamo semplicemente la Pressione moltiplicando per -1 otteniamo l'intervallo $[-1..0]$ che non genera alcuna uscita (L'intervallo valido per il valore di uscita è $[0..1]$). Quindi occorrebbe aggiungere 1 al risultato precedente per riportare il valore del segnale di uscita nell'intervallo $[0..1]$.

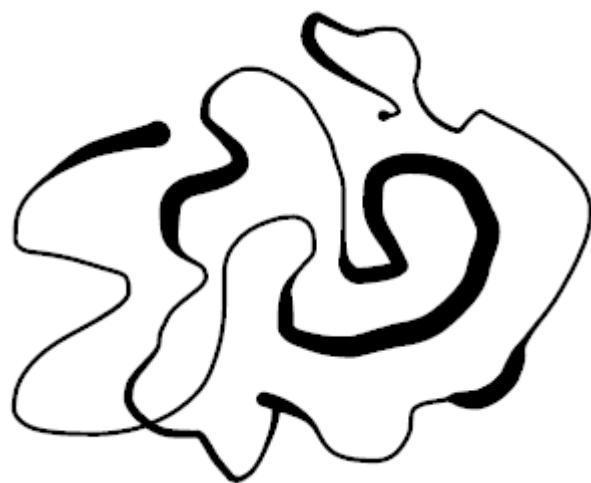
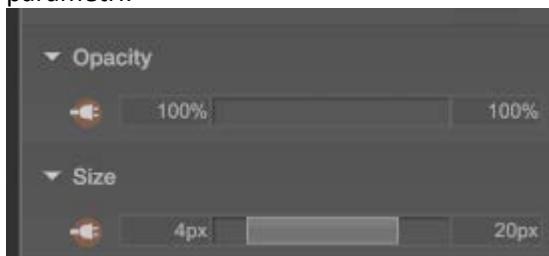
Comunque il modo più semplice per invertire un valore:

Valore Uscita = 1 – Valore Ingresso, cioè Dimensione = 1 – Pressione

Quindi per realizzare il controller utilizziamo il nodo Sub dove in Input B colleghiamo la Pressione e in Input A assegniamo il valore 1 :

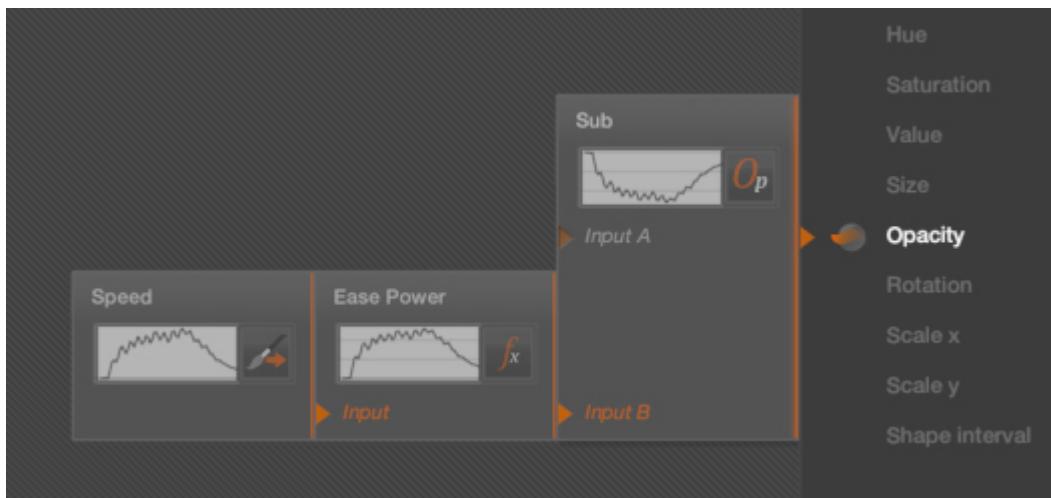


Questo è il risultato quando il controller viene applicato al pennello **Felt pen default** con i seguenti parametri:



La Dimensione del tratto diminuisce con l'aumentare della Pressione.

Provate anche il seguente controller (che è molto simile al precedente):



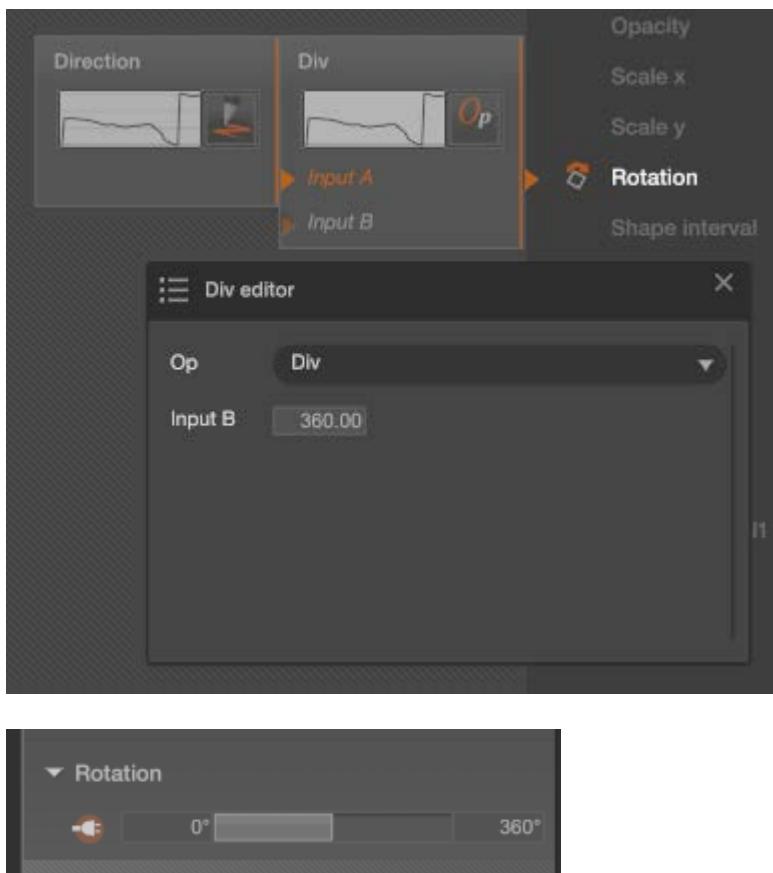
ESEMPIO

Vogliamo che il pennello venga ruotato nella direzione della pennellata.

In ingresso (input) abbiamo la Direzione (Direction) della pennellata e in uscita (output) abbiamo la Rotazione (Rotation) del Pennello.

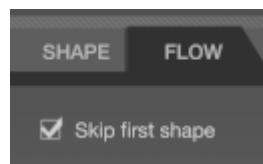
Quindi, prendiamo il nodo **Direction (Direzione)** che restituisce la direzione del tratto da 0 a 2π . Dividiamo per 2π (6.283185307179586) con un nodo **Div (Divisione)** per creare un valore da 0 a 1 da collegare al parametro **Rotation (Rotazione)**.

Poi impostiamo semplicemente il parametro **Rotation (Rotazione)** da 0° a 360° e il pennello ruoterà con la direzione del tratto.

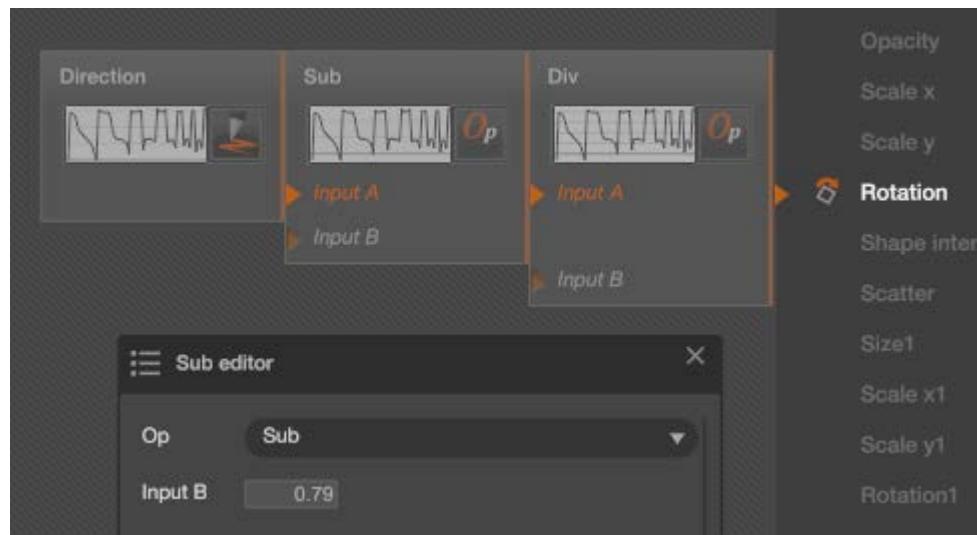




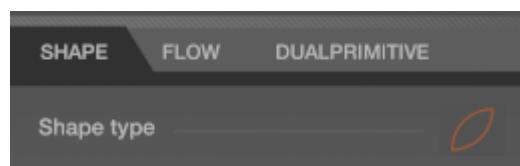
Un consiglio: potresti notare che la prima forma del tuo tratto punta sempre nella stessa direzione (questo è normale poiché non esiste alcuna direzione prima che la seconda forma venga disegnata). Per risolvere il problema esiste l'opzione "Skip first shape" nel pannello **Flow (Flusso)**:

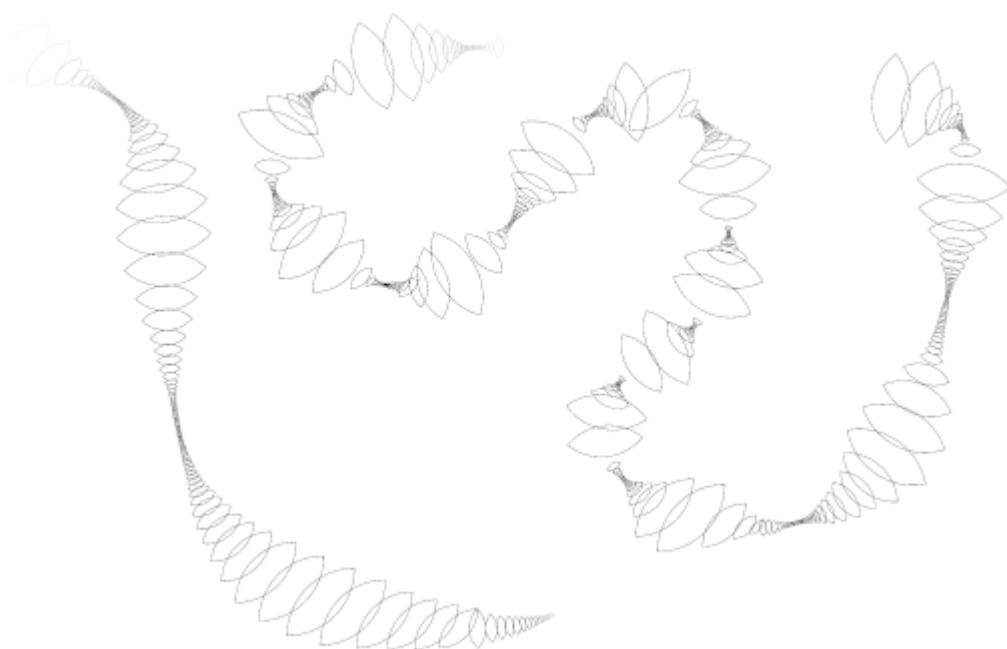


Possiamo migliorare il tratto nel modo seguente:



Sottraiamo $\pi/4$ (0.79) al valore iniziale della direzione per posizionare a 90° la prima forma del pennello:



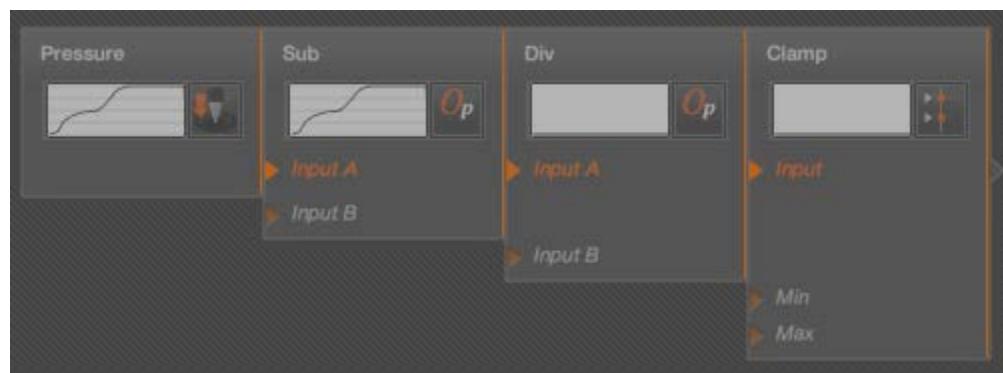


ESEMPIO

Creiamo un controller che mappa l'intervallo di ingresso del segnale da [0,1] ad [a, b].
Dal punto di vista matematico si tratta della seguente funzione:

```
float remap( float value, float minr, float maxr )
{
    return (value-minr) / (maxr-minr);
}
```

Possiamo realizzarla con il seguente controller:



dove:

Sub - Input B = minr
Div - input B = (maxr - minr)
Clamp - Min = 0
Clamp - Max = 1

Quindi puoi collegare l'output a quello che vuoi (al nodo **Easy Power** per esempio).

In sostanza i nodi permettono di applicare al segnale di ingresso delle funzioni matematiche che ci consentono di adattarlo alle nostre esigenze.

Potreste trovare utile il sito <https://www.desmos.com/calculator> che mette a disposizione una calcolatrice grafica con moltissime funzioni.

Brush workshop 5: Esperimenti con un pennello base

Iniziamo questo tutorial con la realizzazione di un pennello di base che servirà come punto di partenza per la costruzione di nuovi pennelli.

Selezioniamo il pennello **Felt pen 4px**:

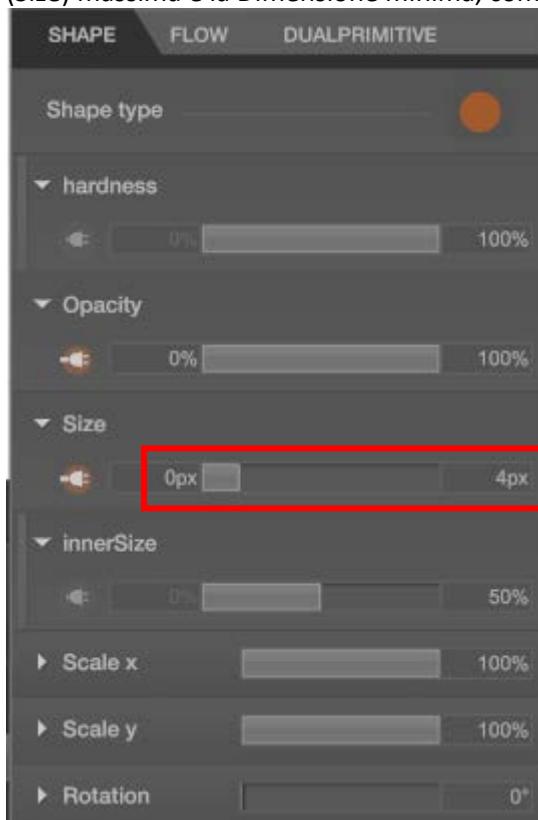


Ecco una pennellata di esempio:

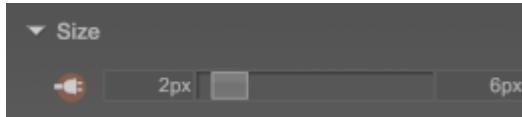


Potete notare che il pennello cambia dimensione e opacità in funzione della pressione della penna.

Il nome deriva dal fatto che simula un pennarello (**Felt pen**) e **4px** sono i pixel di intervallo tra la Dimensione (Size) massima e la Dimensione minima, come si nota dal pannello **SHAPE**:

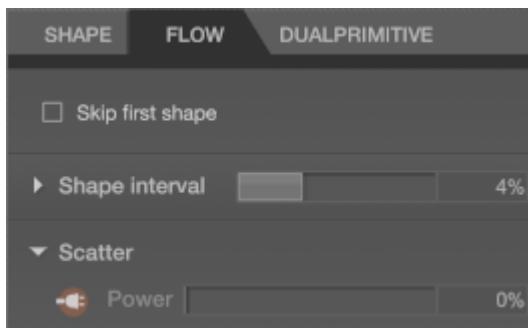


NOTA: La mia tavoletta funziona molto meglio con questo pennello se imposto la Dimensione (Size) minima pari a 2 e la Dimensione massima pari a 6 (oppure 1 e 5 rispettivamente):



Inoltre vi consiglio di modificare la Dimensione utilizzando il cursore che cambia l'intervallo min-max: .

Vediamo il pannello **FLOW**:



Analizzando i pannelli **SHAPE** e **FLOW** notiamo che ci sono tre controller attivi (i controller attivi sono quelli con l'icona “**Spina/Presa**” evidenziata:



Il controller dell'**Opacità (Opacity)**

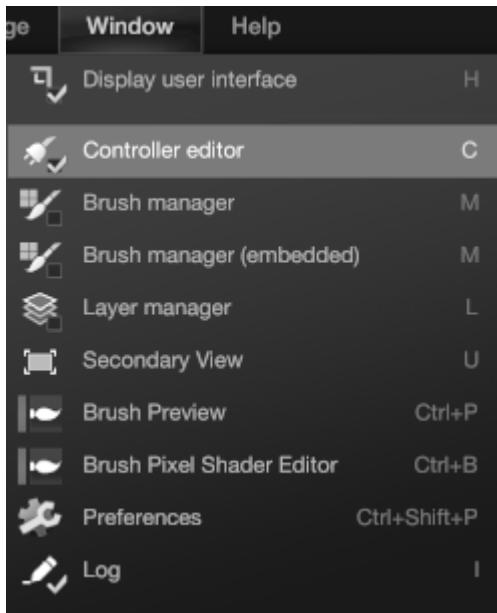


Il controller della **Dimensione (Size)**

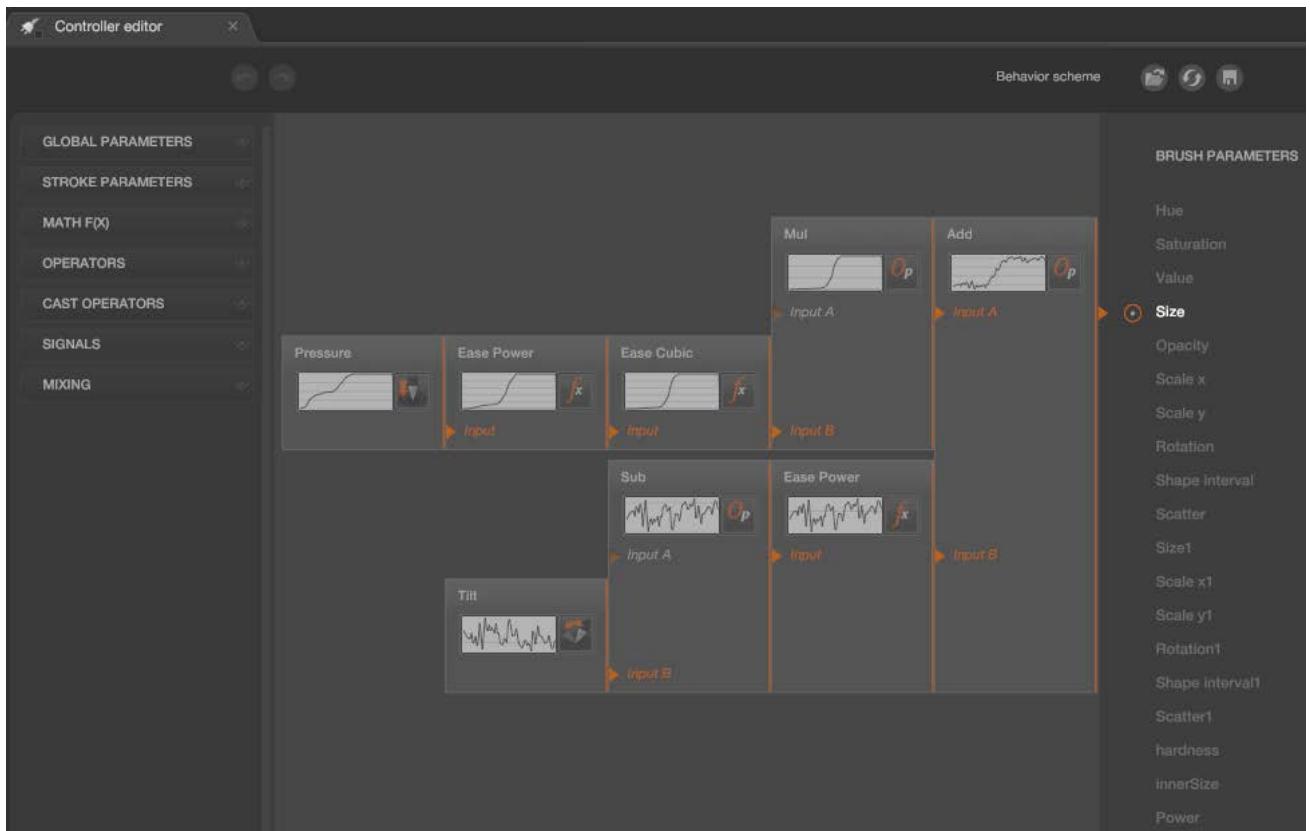


Il controller dello **Scatter (Sparpagliamento)**

Adesso apriamo l'**Editor dei Controller (Controller Editor)** dal menu **Windows->Controller Editor** (oppure premendo il tasto “C”):

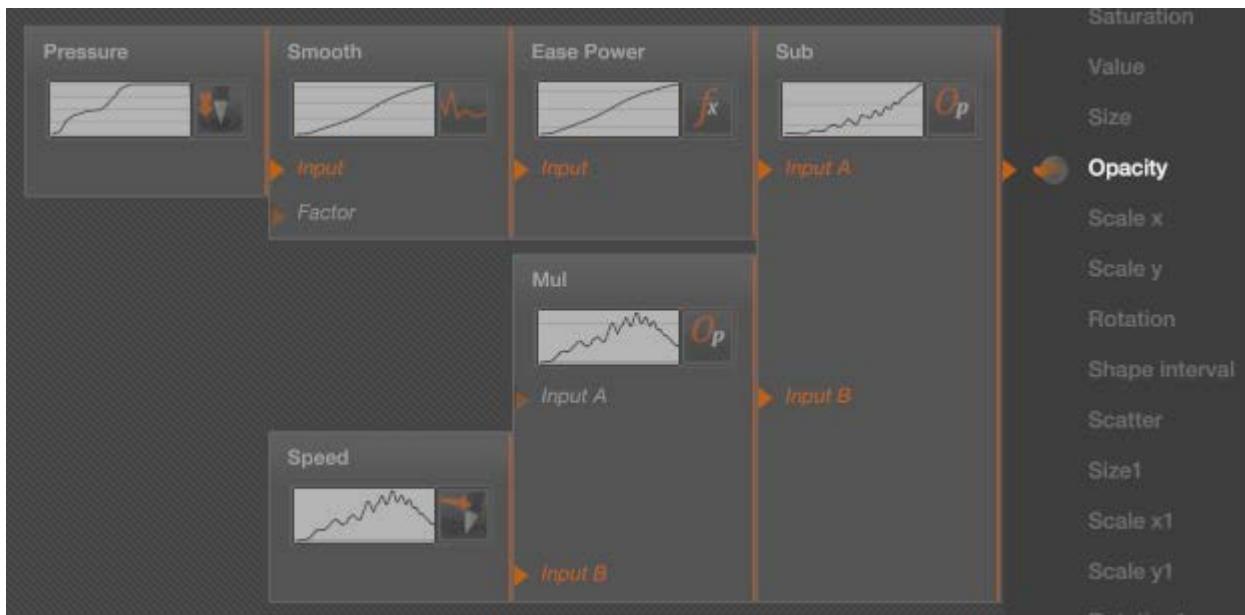


Poi clicchiamo sul parametro **Size** (che si trova a destra nell'Editor nella sezione **BRUSH PARAMETERS**) per visualizzare il relativo controller:

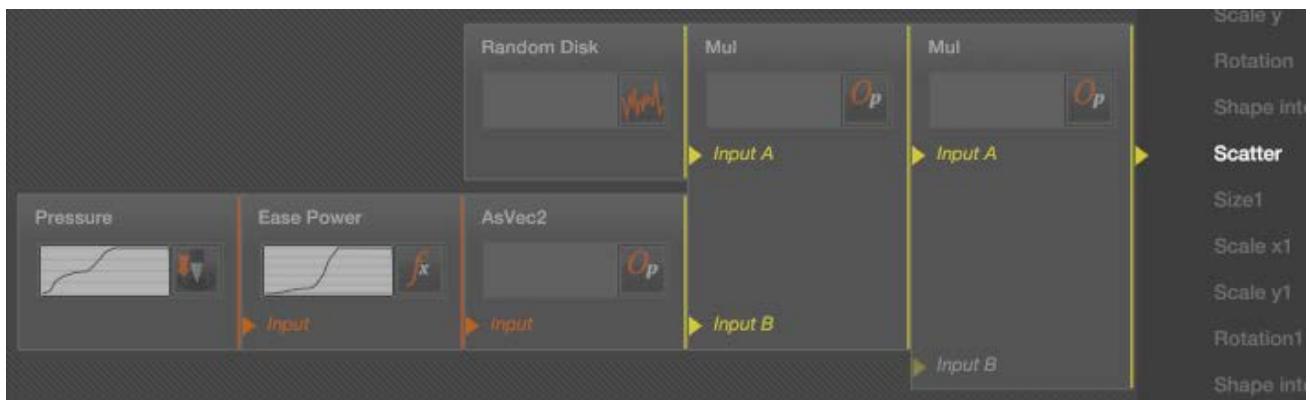


Notiamo che la dimensione dipende dalla **Pressione (Pressure)** e dal **Tilt (Inclinazione)** della penna, quindi se la vostra tavoletta non possiede il parametro Tilt, non sarete in grado di variare la Dimensione con l'Inclinazione (comunque il pennello funzionerà egregiamente).

Adesso clicchiamo sul parametro **Opacity (Opacità)** e visualizziamo il suo controller:



Analogamente per il parametro **Scatter (Sparpagliamento)**:



Adesso creiamo un **Pennello Base** senza alcun controller.

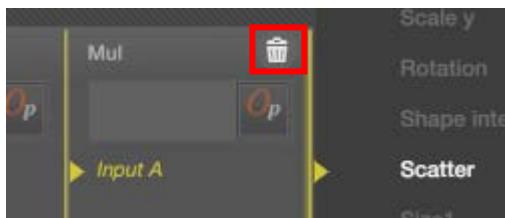
Clicca sui pulsanti evidenziati per eliminare tutti i nodi dei relativi controller (cancellazione a cascata di tutti i nodi precedenti):



Per eliminare il controller del parametro **Size**



Per eliminare il controller del parametro **Opacity**



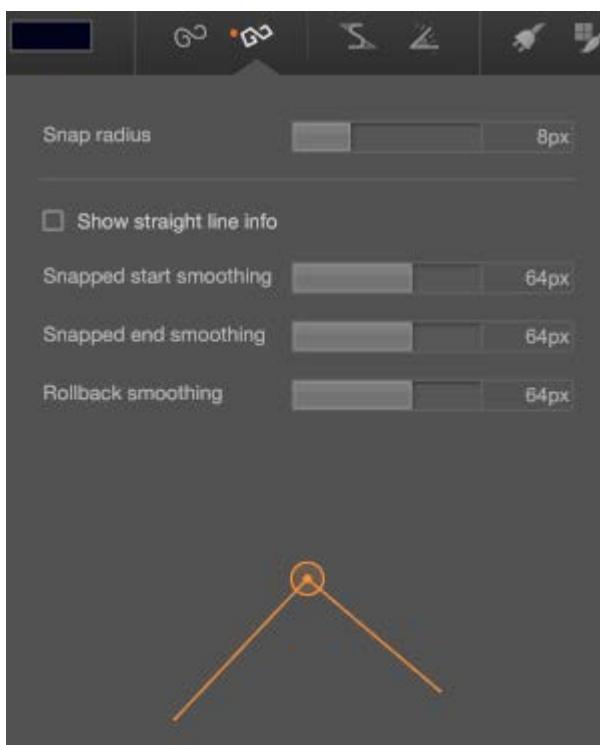
Questa è un pennellata di esempio di questo nuovo pennello:



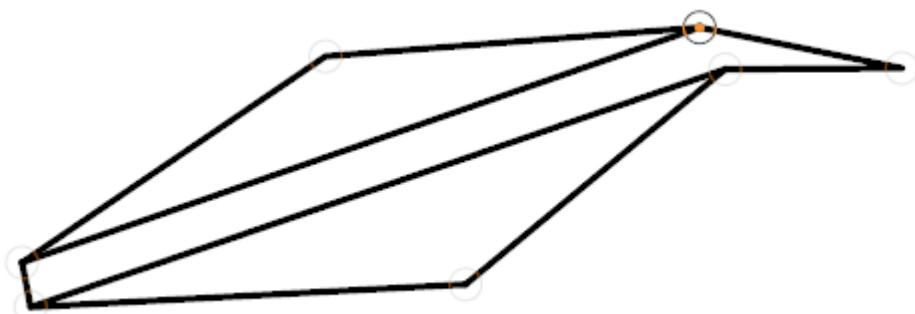
Si tratta di un pennello che non cambia dimensione o opacità durante la pennellata.

A cosa può servire un pennello come questo ?

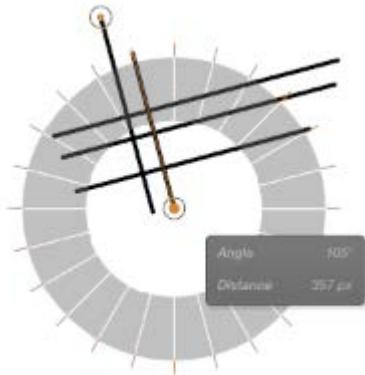
Innanzitutto può essere usato per disegnare linee con tratto costante utilizzando la modalità Straight Line Draw Mode:



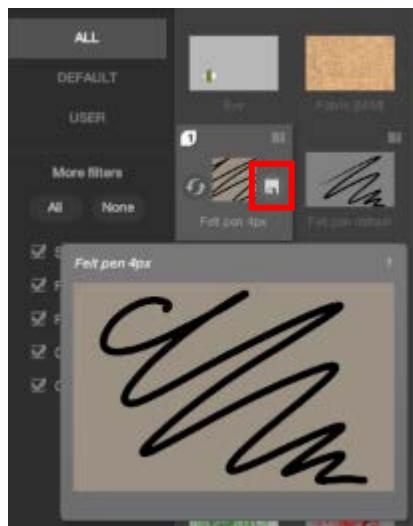
Per ottenere questi risultati:



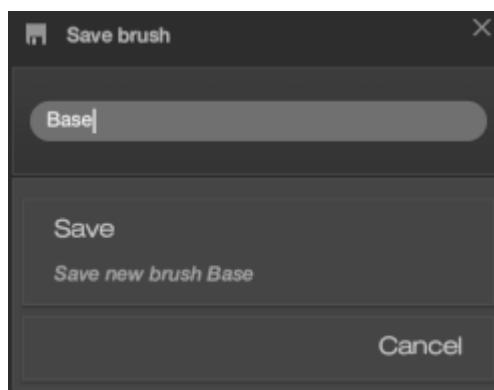
Oppure, quando siamo in modalità a mano libera (**Free Hand Mode**) possiamo disegnare con linee rette premendo il tasto il tasto “Shift”:



Per salvare il nostro pennello base dobbiamo aprire il **Gestore dei Pennelli (Brush Manager)** dal menu **Windows->Brush Manager** (oppure premendo il tasto “M”) e cliccare sull’icona evidenziata:



Scrivere il nuovo nome e premere il pulsante “Save”:

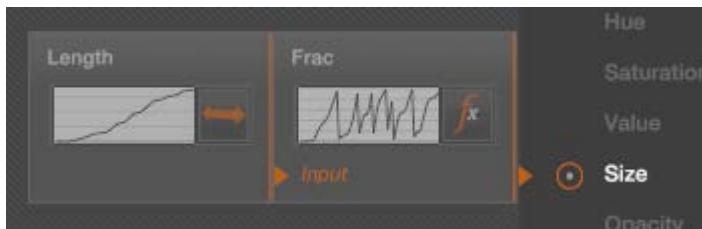


NOTA: Non è possibile sovrascrivere i pennelli predefiniti (default), quindi potete modificare qualunque parametro e vedere i relativi effetti senza preoccuparsi di sovrascrivere qualcosa.

Cominciamo a fare alcuni esperimenti con i controller:

Pennello che modifica la Dimensione (Size) in maniera casuale durante la pennellata

Prendiamo il nodo **Stroke Parameters->Length** e lo attacchiamo al nodo **Math>Frac**, poi connettiamo il tutto al parametro **Size**:



Poi poniamo la **Dimensione (Size)** a 1..20:



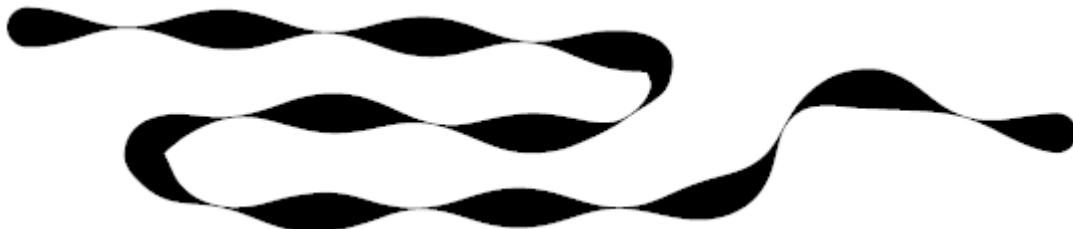
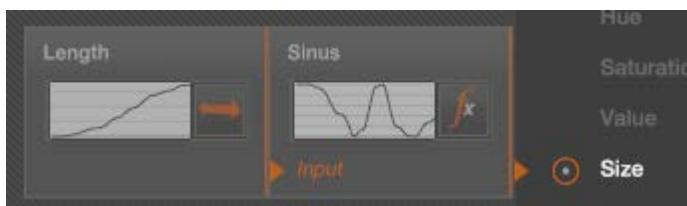
Tracciamo alcune pennellate:



Il tratto non è proprio “casuale”, ma il risultato è interessante.

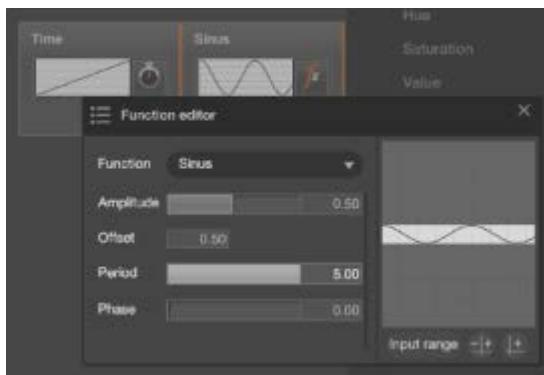
La **Distanza (Length)** aumenta sempre e il nodo **Frac** ne prende la parte frazionaria, quindi la sua uscita è sempre un numero compreso tra 0 e 1 che cresce linearmente.

Adesso sostituiamo il nodo **Frac** con il nodo **Math->Sinus** e tracciamo una pennellata:



Un pò più casuale, ma periodico (perchè il Seno restituisce valori periodici compresi tra 0 e 1).

Proviamo a sostituire il nodo **Length** con il nodo **Stroke Parameters->Length** e impostiamo il periodo del nodo **Sinus** a 5.0:



Ci siamo ! Il nodo **Time** aumenta come il nodo **Length**, ma, poichè le pennellate vengono fatte con tempi diversi, otteniamo un andamento più casuale.

Pennello che disegna gruppi di punti sparsi di dimensione casuale



Partendo dal pennello Base dobbiamo modificare il parametro **Shape Interval** del pannello **Flow** e creare un controller per il parametro **Scatter (Sparpagliamento)**.

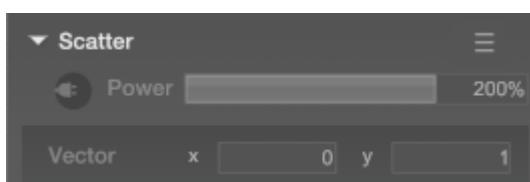
Impostiamo i seguenti parametri:



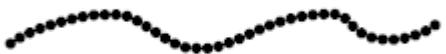
Size min = 8 e Size max = 22



Shape Interval = 50%



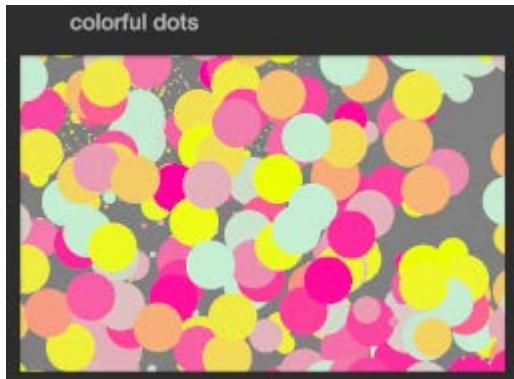
Scatter = 200%



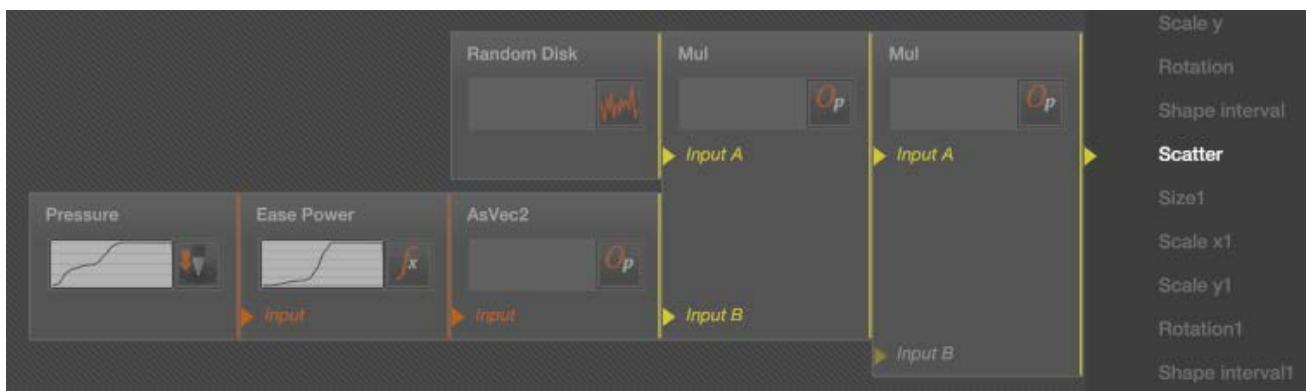
Pennello attuale

Siccome non abbiamo idea di come creare un controller per lo Scatter...lo copiamo da un pennello che lo implementa ☺.

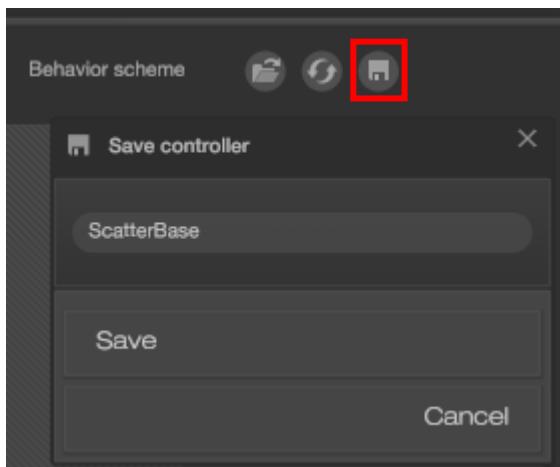
Selezioniamo il pennello **colorful dots**:



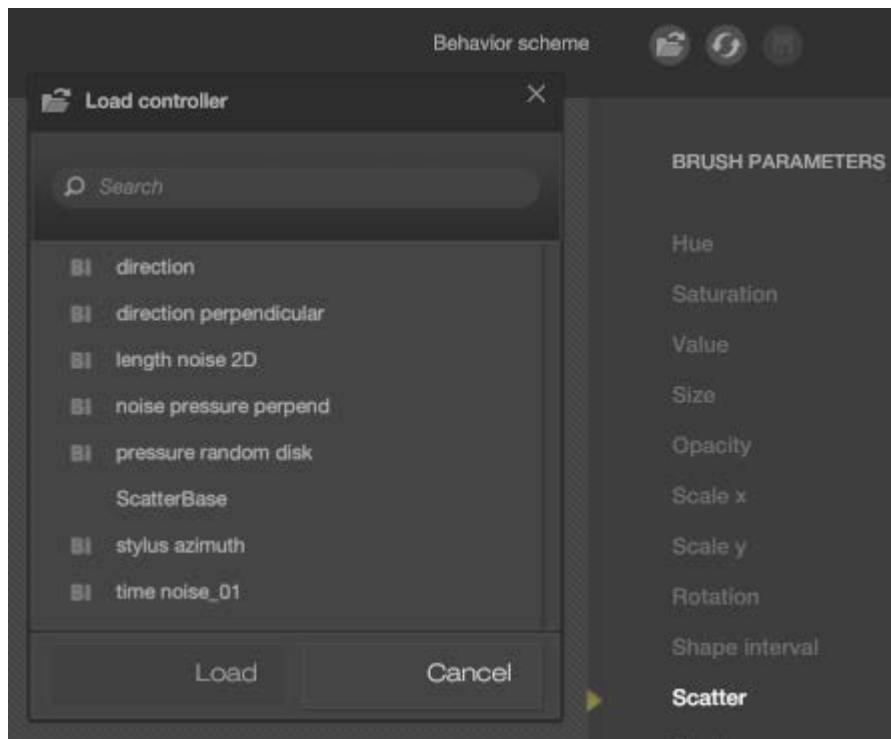
Apriamo il **Brush Manager** e selezioniamo il parametro **Scatter**:



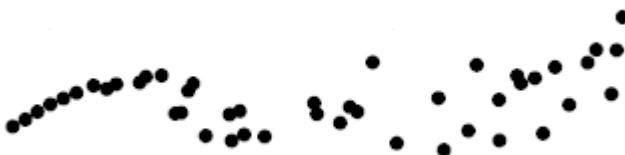
Salviamo questo controller con il nome **ScatterBase**:



Selezioniamo nuovamente il nostro pennello **Base** e carichiamo sul parametro **Scatter** il controller appena salvato (ScatterBase):

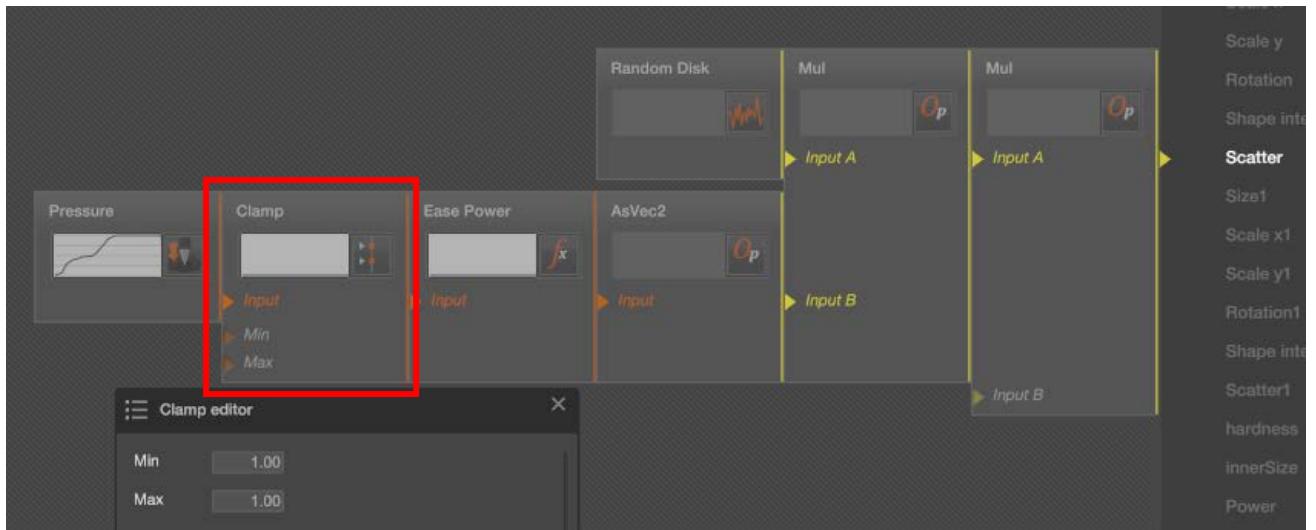


Il nostro pennello dovrebbe avere il seguente comportamento:



Siccome il valore dello Scatter è proporzionale alla Pressione, all'inizio del tratto (con la Pressione bassa) abbiamo un valore quasi nullo per lo Scatter (quasi nessun sparpagliamento dei punti).

Per rendere costante il valore della Pressione utilizziamo il nodo **Mixing->Clamp** con i valori Min = 1 e Max = 1:



In questo modo il valore all'uscita del nodo **Clamp** vale sempre 1 ed otteniamo il seguente tratto:



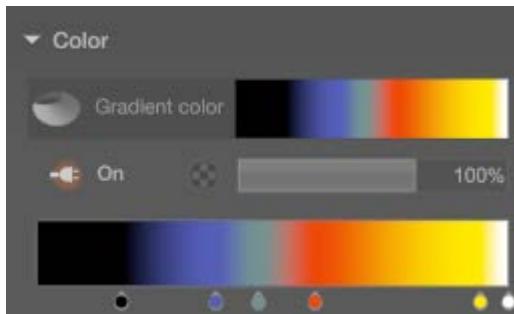
Ci siamo quasi: adesso dobbiamo modificare la dimensione dei punti. Per fare questo creiamo il seguente controller per il parametro **Size (Dimensione)**:



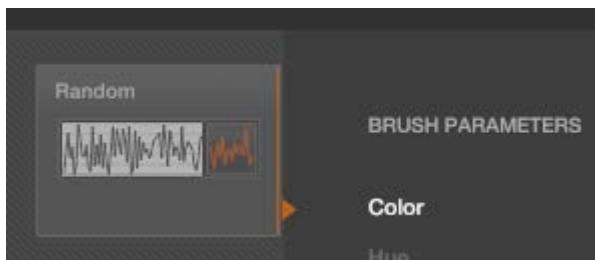
Per ottenere il risultato finale del nostro pennello:



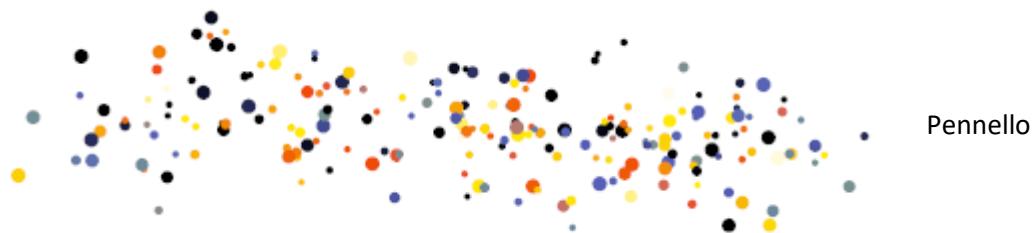
Per ottenere dei punti colorati in maniera casuale possiamo impostare il colore su **Gradient Color**, creare un gradiente ed utilizzare un controller per il parametro **Color**:



Gradiente



Controller



Pennello che crea pennellate tratteggiate

Il nostro pennello Base ha questo tratto:



Per ottenere una linea tratteggiata occorre agire sull'Opacità del pennello.

Dobbiamo creare un controller che abbia in uscita un segnale del seguente tipo:



I nodi che ci servono sono i seguenti:

Mod (Float)

Modulo. Resto della divisione tra l'input A e l'input B.

In termini pratici, consente di ripetere un valore da zero a un limite definito. Ad esempio, la lunghezza con un modulo a 0.5 aumenterà come al solito, ma quando raggiunge 0,5, verrà riavviata da 0.

Sub (Float)

Sottrae l'input B all'input A.

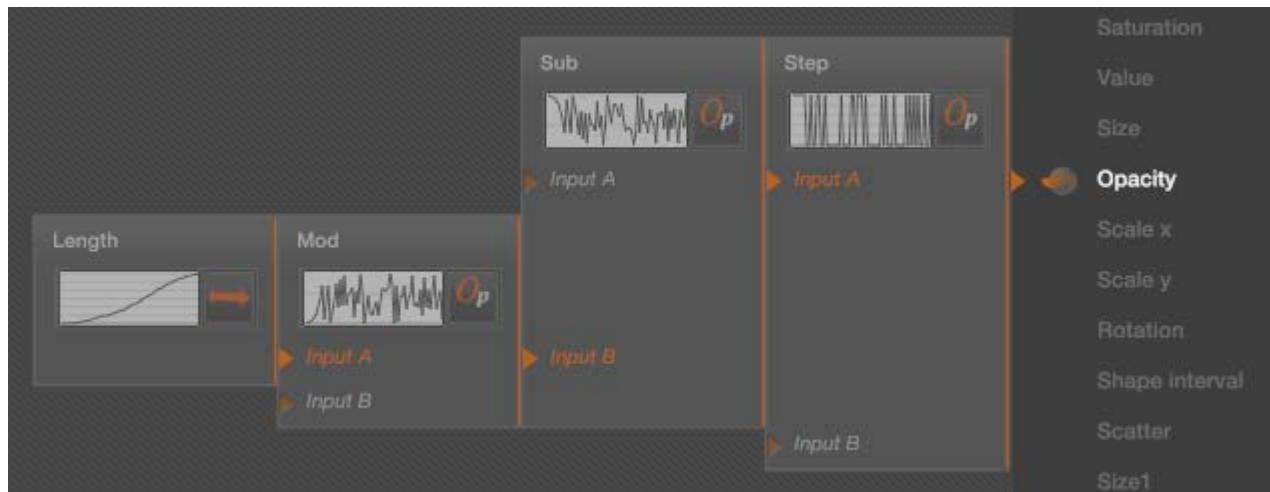
Utile per invertire i valori: collegare il valore di input in B e impostare A a 1. Ad esempio, se la Pressione controlla l'Opacità, allora si otterrà un tratto più opaco quando si preme più forte e 1-Pressione farà l'opposto, rendendolo più opaco quando si preme di meno.

Step (Passo)

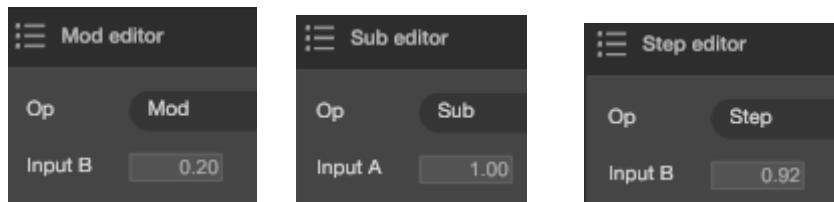
Restituisce 0 se l'ingresso A è inferiore a B, 1 se superiore.

In altri termini, il valore B viene utilizzato per scegliere se l'ingresso A viene convertito in 0 o in 1. Ad esempio: Se B è impostato su 0.5, allora 0.1 restituisce 0, 0.4 restituisce 0, 0.6 restituisce 1, 0.9 restituisce 1.

Costruiamo il seguente controller e colleghiamolo al parametro **Opacity**:



I parametri dei nodi sono i seguenti:



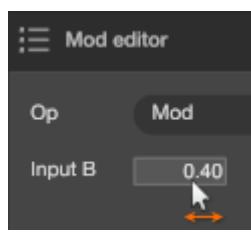
Come funziona ?

Il segnale che esce dal nodo Mod determina la lunghezza della linea vuota (opacità al 0%), ma varia da 0 al valore di Input B, quindi viene invertito dal nodo Sub. Il nodo Step imposta l'uscita a 0 oppure a 1 confrontando il valore Input A con il valore di Input B (valore di soglia), in altre parole questo nodo controlla la lunghezza della linea vuota (Opacità al 100%).

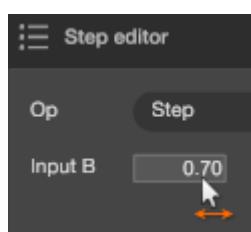
Per verificare questo comportamento aprite la finestra **Brush Preview** (**Windows-> Brush Preview** o con "Ctrl-P") e disegnate qualche tratto:



Provate a modificare i seguenti parametri:



Varia la lunghezza della linea vuota (Opacità 0%)



Varia la lunghezza della linea piena (Opacità 100%)

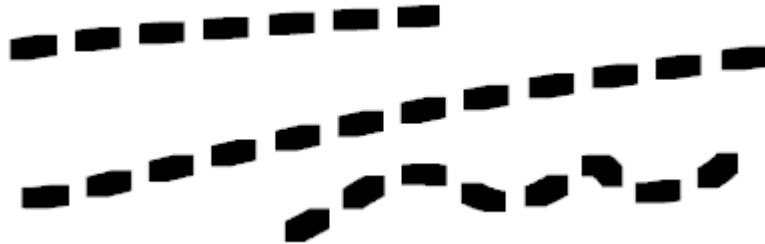
Il nostro pennello si presenta in questo modo:



Se vogliamo una linea tratteggiata non arrotondata possiamo cambiare la forma (**Shape**) del pennello, scegliendo la forma quadrata piena:

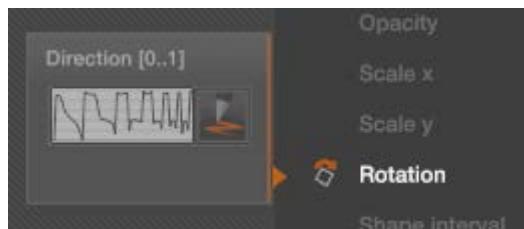


Proviamo il pennello:



Notiamo che nei tratti curvi il pennello non si comporta correttamente, per risolvere questo problema dobbiamo far ruotare la punta del pennello con la direzione della penna.

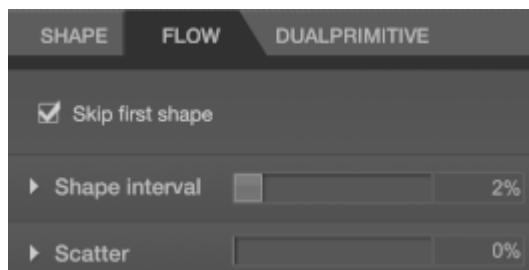
Creiamo il seguente controller:



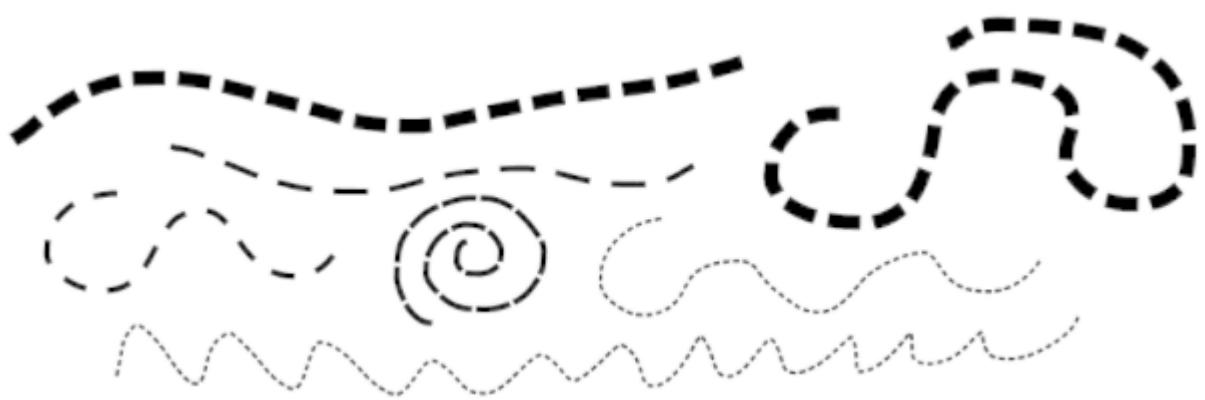
E modifichiamo l'intervallo del parametro **Rotation (Rotazione)** da 0 a 360 gradi.



Siccome la prima forma (first shape) non ha una direzione abilitiamo **Skip first shape** che ci permette di non disegnarla:



Ed ecco il risultato finale:



Brush workshop 6: Il Pennello Tubo

In questo tutorial realizzeremo un pennello che permette di disegnare forme simili a dei tubi ondulati (Ripples Pipes):



Impostate i seguenti parametri:

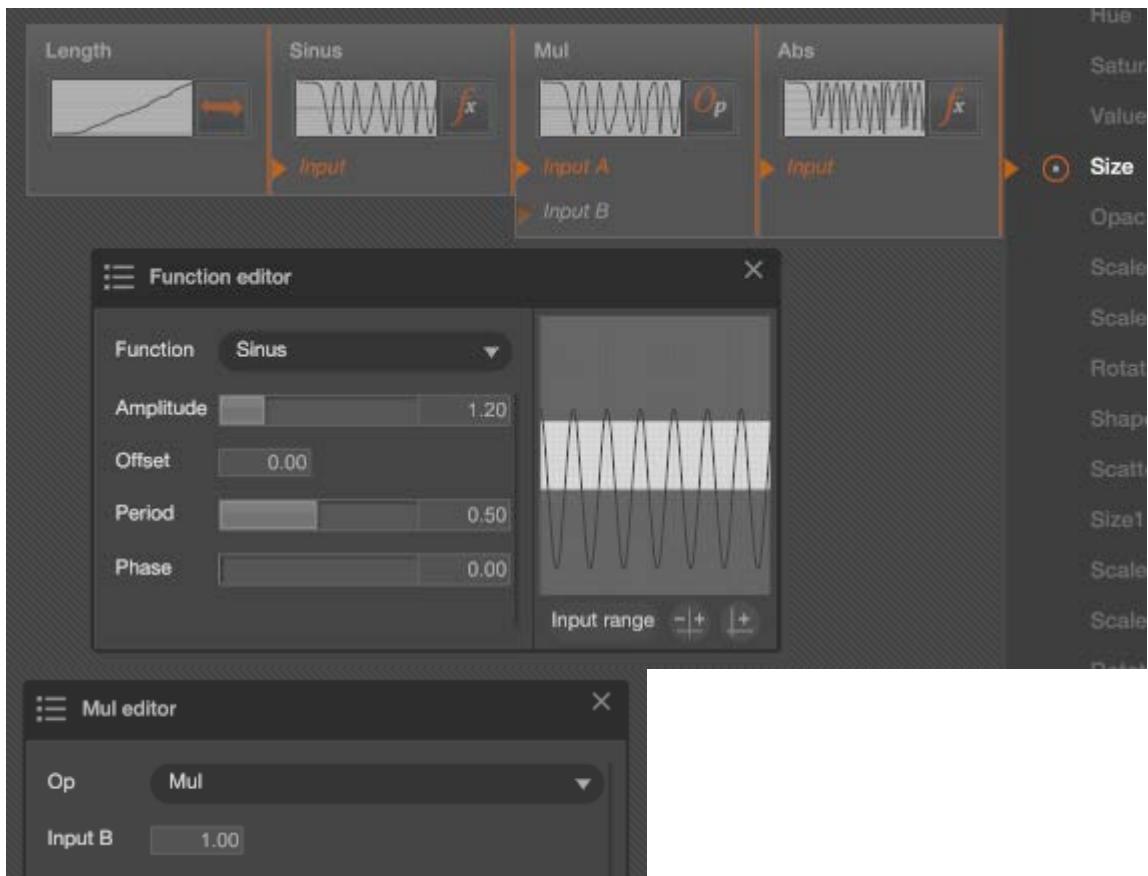
CONTROLLER

Colore (Color)

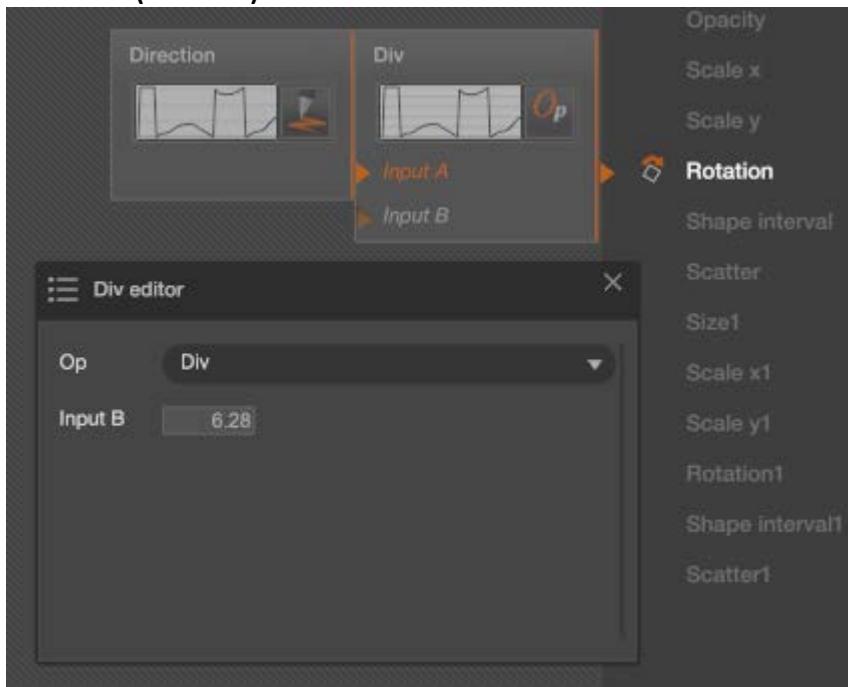
The screenshot shows the Brush Parameters panel with the following settings:

- Color:** Hue, Saturation, Value, Size, Opacity, Scale x, Scale y, Rotation, Shape interval.
- Function editor:** Function: Sinus, Amplitude: 1.00, Offset: 0.00, Period: 0.50, Phase: 0.00.
- Length:** Sinus function.
- Abs:** Abs function.

Dimensione (Size)

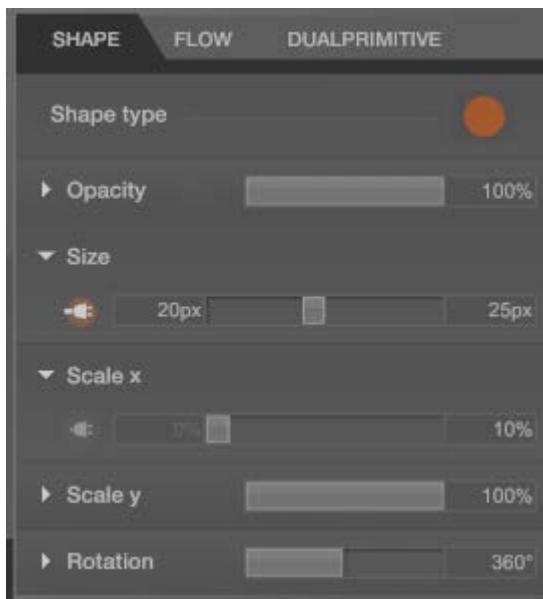


Rotazione (Rotation)



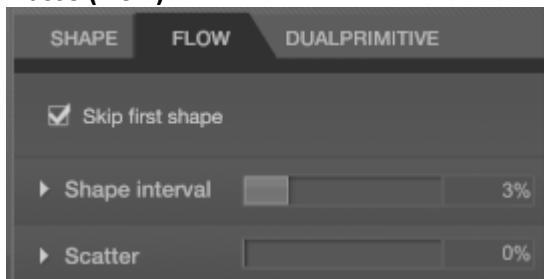
PARAMETRI del PENNELLO

Forma (Shape)

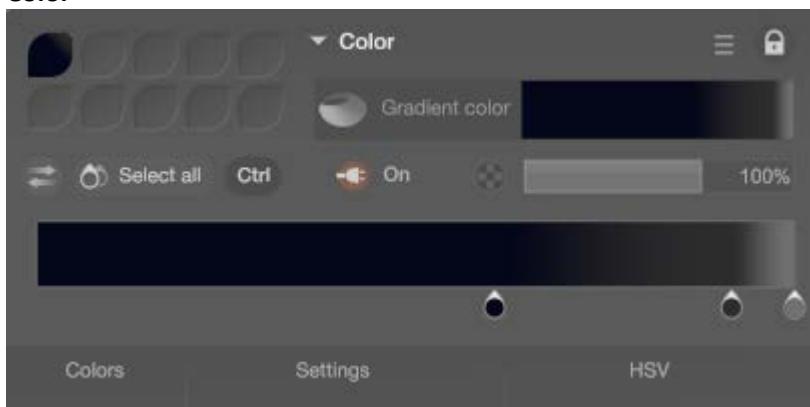


La **Dimensione (Size)** minima e massima e il loro intervallo modificano fortemente il pennello.

Flusso (Flow)



Color



Adesso per capire come funziona ogni parte provate a modificare i vari parametri per vedere gli effetti sul pennello. Se avete seguito i tutorial precedenti non dovrebbe essere troppo difficile 😊.

Breve introduzione al Brush Shading Language (BSL)

Introdurremo un nuovo linguaggio di shading (sfumatura) personalizzato: il BSL (Brush Shading Language).

Viene utilizzato per controllare come i pixel vengono inseriti nella tela. È principalmente un superset del linguaggio [HLSL](#) che introduce molti aiuti e semplificazioni.

Attualmente, i pennelli di BlackInk utilizzano solo un colore per primitiva e i Controller modificano solo la dimensione, la rotazione, ecc., ma non sono in grado di modificare il modo in cui il colore viene applicato all'intera primitiva. Il linguaggio BSL permette proprio questo e apre un intero nuovo mondo di possibilità (infinite).

In breve, sarà possibile programmare effetti come: smudge blurs, smoothed primitives, texture masking, noises, color aberration, water effects e molto altro ancora ...

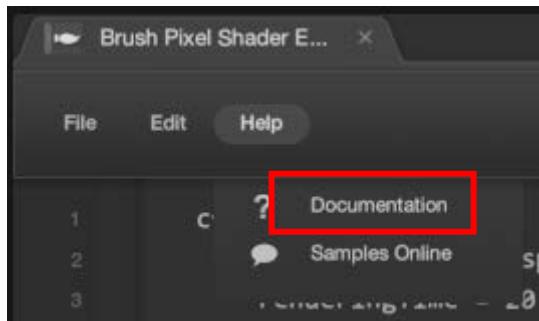
Questo è davvero interessante, soprattutto per gli utenti orientati alla programmazione. Gli artisti puri non devono preoccuparsi, tutta questa complessità sarà nascosta e sarete in grado di utilizzare effetti personalizzati senza dover sapere come sono stati fatti.

La versione 0.357 include solo un editor di testo per scrivere codice in BSL, perché uno strumento basato su grafi (nodi) richiederà molto più tempo per essere sviluppato (Questo funzionalità è molto più complessa del sistema dei Controller).

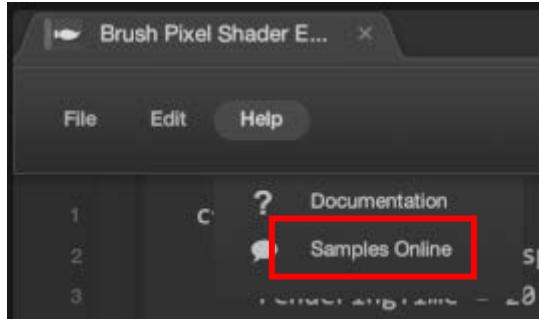
Un sacco di codice di Black Ink è stato riscritto per supportare questa nuova funzione, quindi dovremo aspettare un pò per avere la versione finale, ma ne varrà la pena.

Possiamo trovare maggiori informazioni nei seguenti documenti (**menu Help del Brush Pixel Shader Editor**):

BSL Reference (Manuale di riferimento)



BSL Samples (Esempi dal manuale di riferimento)



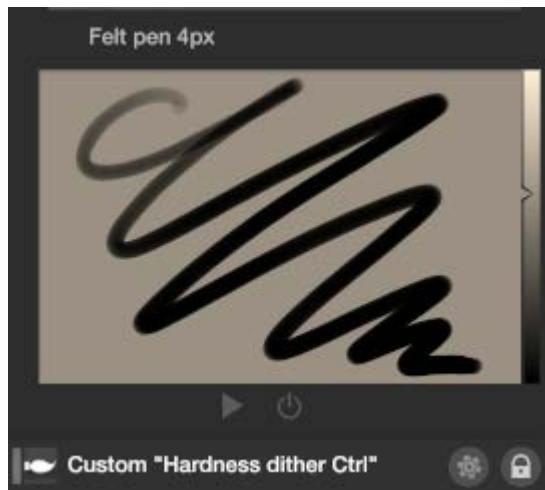
BSL Code (Codice di tutti gli shader BSL presenti in BlackInk 0.357)

(vedi documento allegato)

Utilizzo del Brush Shader Editor

Per iniziare vediamo come possiamo utilizzare questi shader BSL con BlackInk.

Applicheremo i vari script BSL al pennello **Felt Pen 4px** (con **Size min = 6** e **Size max = 10**), ma vi consiglio di sperimentarli anche con altri pennelli:



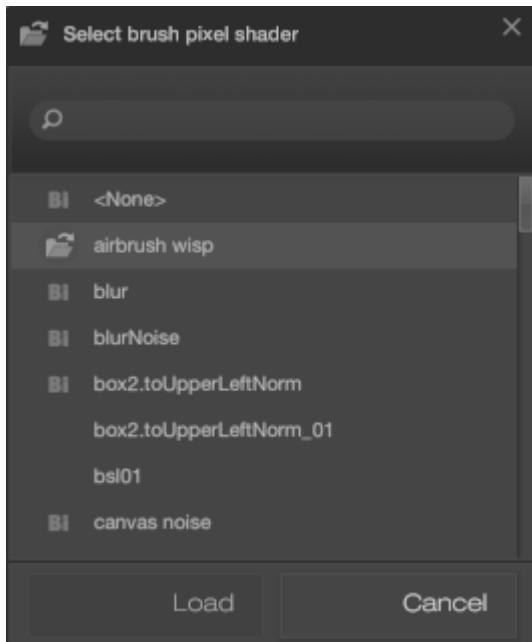
Dalla figura notiamo che questo pennello usa lo shader **“Hardness dither Ctrl”**.

Possiamo modificare questo shader oppure possiamo caricarne uno nuovo:

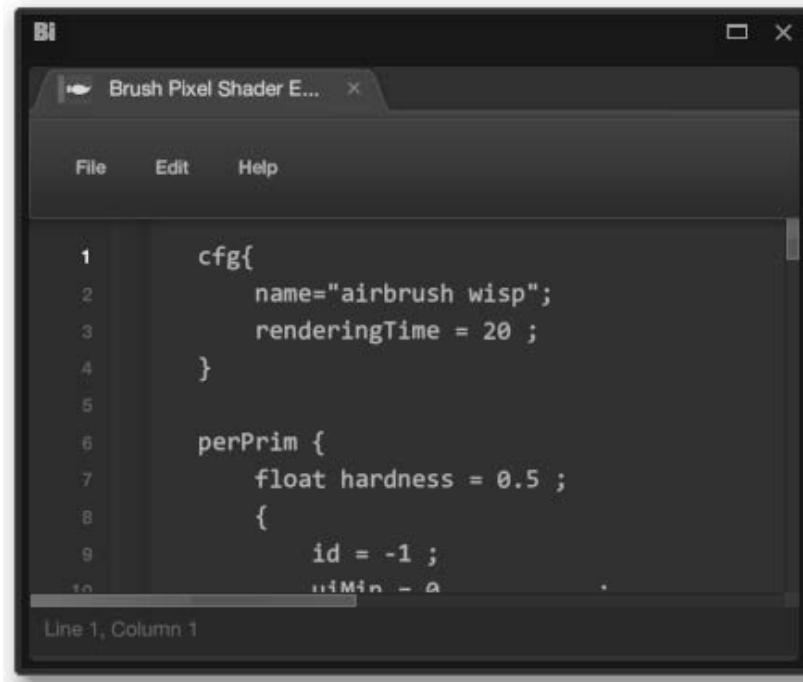
Il pulsante apre la finestra del **Brush Pixel Shader Editor** (editor di testo):

```
1     cfg{name="Hardness dither Ctrl";}
2
3     perPrim {
4         float hardness = 0.5 ;
5         {
6             id = -1 ;
7             uiMin = 0 ;
8             uiMax = 1 ;
9             uiTab = "shape" ;
10            uiFormat = percent ;
11        }
12    }
```

Il pulsante permette di caricare un shader esistente in sostituzione di quello attivo:



Proviamo a selezionare lo shader **airbrush wisp** e poi caricarlo premendo il pulsante **Load**:



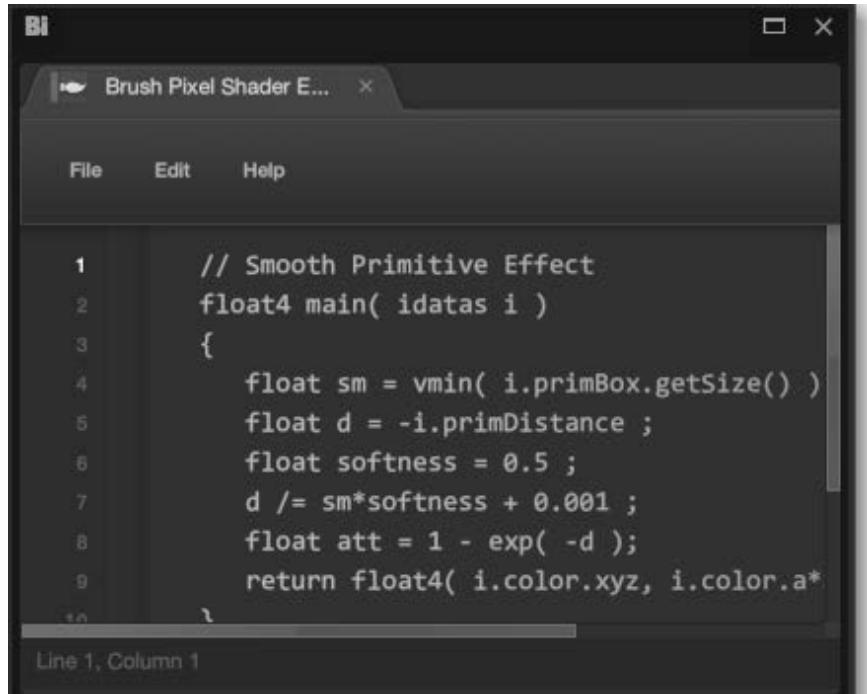
Notiamo che il pennello ha un comportamento molto diverso da quello precedente: questo è dovuto al nuovo shader.

Nota: La visualizzazione reale dei tratti del pennello avviene quando il disegno ha lo zoom 1:1 (**Ctrl Num 1**).

Adesso proviamo ad inserire un nuovo shader (il cui codice è preso dagli esempi forniti sul forum di BlackInk). Selezioniamo tutto il codice dell'editor (**Ctrl-A**) e lo eliminiamo (**Del**), poi inseriamo il seguente shader (BSL01):

```
// Smooth Primitive effect  
float4 main( idatas i )  
{
```

```
float sm = vmin( i.primBox.getSize() ) ;
float d = -i.primDistance ;
float softness = 0.5 ;
d /= sm*softness + 0.001 ;
float att = 1 - exp( -d );
return float4( i.color.xyz, i.color.a*att ) ;
}
```



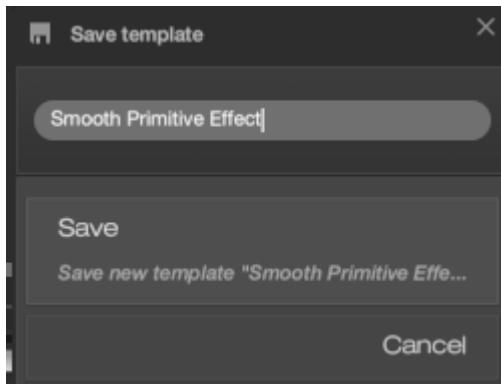
The screenshot shows a window titled "Brush Pixel Shader E...". The menu bar includes "File", "Edit", and "Help". The main area contains the following shader code:

```
1 // Smooth Primitive Effect
2 float4 main( idatas i )
3 {
4     float sm = vmin( i.primBox.getSize() ) ;
5     float d = -i.primDistance ;
6     float softness = 0.5 ;
7     d /= sm*softness + 0.001 ;
8     float att = 1 - exp( -d );
9     return float4( i.color.xyz, i.color.a*
10 }
```

Line 1, Column 1



Possiamo anche salvare questo shader tramite il menu dell'editor **File->Save**:

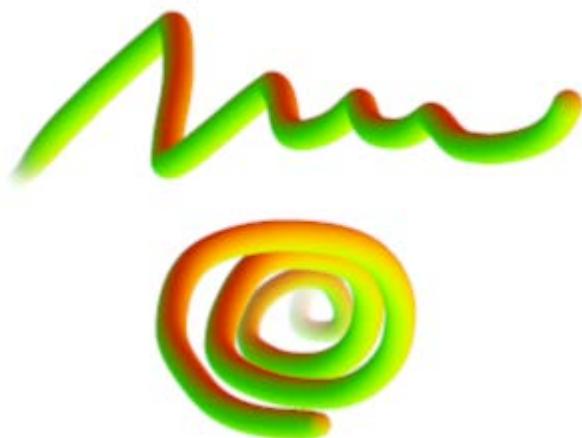


Nota: Non è possibile sovrascrivere gli shader predefiniti, quindi potete sperimentare con tutta tranquillità.

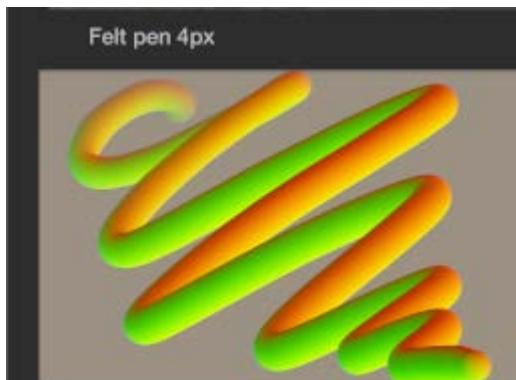
Proviamo un altro shader (BSL02):

```
// Primitive space display
float4 main( idatas i )
{
    matrix2 t = i.primBox.getToUpperLeftTransfo() ;
    float2 p = t.transform( i.pos );
    p /= i.primBox.getSize();

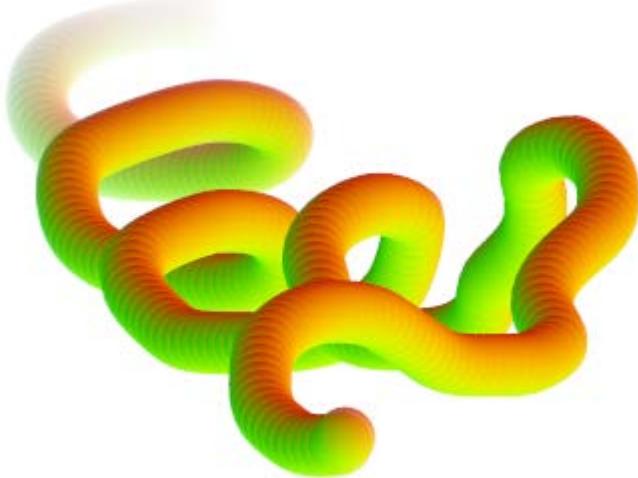
    return float4( p.x, p.y, 0, i.color.a ) ;
}
```



La finestra di Anteprima del Pennello mostra in tempo reale il risultato dello shader (se non ci sono errori):



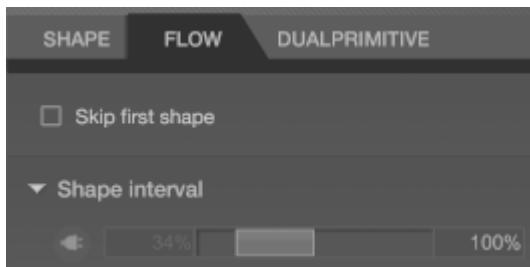
Provate a cambiare i parametri del pennello (**Size max= 18** e **Shape interval = 14%**):



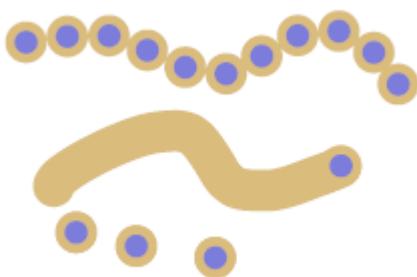
Un altro esempio (BSL03):

```
// Distance to primitive effect
float4 main( idatas i )
{
    float d = toDrawSpace( i.primDistance ) ;
    float l = abs( cos( d*0.12 ) ) > 0.85 ? 1 : 0 ;
    float3 col = lerp( float3(0.2,0.2,0.7), float3(0.7,0.5,0.2), l );
    return float4( col, 1 );
}
```

Modifichiamo lo “**Shape interval**”:



Questi sono alcuni tratti del pennello:



Analisi di uno shader (hardness ctrl)

BlackInk possiede un BSL predefinito di nome “**hardness ctrl**”.

Il codice BSL riportato sotto è una versione modificata in cui ho aggiunto alcuni commenti e piccole modifiche per rendere il codice più chiaro (**Hardness Control**):

Hardness Control



```
// configurazione
cfg{name="Hardness Control";} // nome dello shader

// Definizione dei parametri e relativa interfaccia grafica
// Questi parametri vengono aggiunti automaticamente alla struttura idatas

perPrim {
    float hardness = 0.5; // parametro "hardness"
    {
        id = 1;           // numero identificativo del parametro
        uiMin = 0;         // valore minimo del parametro
        uiMax = 1;         // valore massimo del parametro
        uiFormat = percent; // unità di misura del parametro
        uiTab = "BSL";     // nome della Pannello (nuovo o esistente)
        uiName = "hardness"; // nome del parametro
    }
    float innerSize = 0.5; // parametro "innerSize"
    {
        id = 3;           // numero identificativo del parametro
        uiMin = 0;         // valore minimo del parametro
        uiMax = 1;         // valore massimo del parametro
        uiFormat = percent; // unità di misura del parametro
        uiTab = "BSL";     // nome del Pannello (nuovo o esistente)
        uiName = "inner Size"; // nome del parametro
    }
}

// compute the normalized distance from the edge to the center of the primitive
float cptPrimNormDist( idatas i )
{
    // retreive maximum primitive axe size
    float sm = 0.5*vmax( i.primBox.getSize() );
    // compute normalized current distance to the primitive edge
    float d = max( -i.primDistance, 0 ) / sm;

    return i.primDistanceValid ? d : 0;
}
```

```

}

// compute Hardness
// d - distance from primitive edge [0,1]
// h - hardness parameter [0,1]
// i - distance to the inner "safe" size [0,1]
//
float cptHardness( float d, float h, float i )
{
    float attsize = (1-i)*2*(1-h) ;
    float att = saturate( (d - (1-i - attsize*0.5) ) / max( attsize,0.0001) ) ;
    return smoothLerp(0,1,att);
}

float4 main( idatas i )
{
    float d = cptPrimNormDist( i );

    float att = cptHardness( d, i.hardness, i.innerSize );
    att = i.primDistanceValid ? att : 1 ;

    float alpha = saturate(i.color.a*att) ;
    float3 col = i.color.xyz ;

    return float4( col, alpha ) ;
}

```

Notare i nuovi parametri del pennello messi a disposizione da questo BSL (**hardness** e **innerSize**) nel pannello **Shape**:



La funzione main e la struttura idatas

Ogni BSL deve avere una funzione "**main**" che restituisce un **colore ARGB** (di tipo **float4**)
Esempio minimale:

```
float4 main()
{
    return 1 ;
}
```

In genere la funzione main ha una struttura di ingresso chiamata "**iDatas**":

Esempio minimale:

```
float4 main( idatas i )
{
    return i.color;
}
```

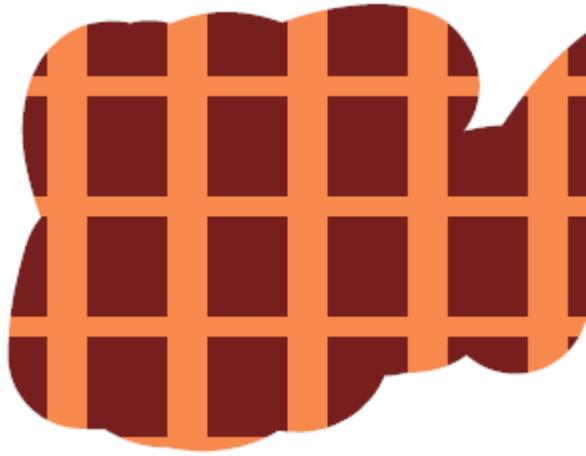
Vediamo quali sono i parametri della struttura predefinita **iDatas**:

iDatas		
float4	color	Returns primitive data for currently selected color. <i>Note : if the BSL doesn't use this parameter at all default Brush color parameters will be removed from the UI</i>
float2	pos	Returns the current canvas pixel position.
float2	strokePos	Returns pixel position of the current stroke.
float2	strokePrecedPos	Returns pixel position of the precedent stroke.
float2	strokeStartPos	Returns pixel position at the exact beginning of the stroke.
float	time	Returns stroke duration in seconds.
float	dist	Returns stroke length in pixels on the canvas.
float	nbUserStroke	Returns the number of stroke the user did since the start of Black Ink
bool	eraser	Is the Eraser mode activated for the brush ?

float	primDistance	The distance signed in pixel to the primitive (a negative value indicates that one is in the primitive) in the space of the canvas. <i>Note : Some Brush Engines wil not return correct value. It may be necessary to check with primDistanceValid.</i>
bool	primDistanceValid	Boolean check for validity of returned value from primDistance.
float2	primUV	The current UV normalized position in the primitive. The UV position is typically defined in the range [0,1]
<u>box2</u>	primBox	Returns Box primitive on the canvas.
<u>box2</u>	primBox1	Returns Box primitive on the canvas generated from an auxiliary system
float	primShapeId1	Returns isEmpty the ID of the generated primBox1 since the start of the stroke
bool	primBox1Valid	Boolean check for validity of returned value primBox1. Because some Brush engine doesn't generate an auxiliary primitive
..	All data from perPrim	All data defined by perPrim will be added to the idatas structure automatically.

Il seguente BSL disegna una griglia calcolando il colore in base alla posizione del pennello nel disegno:

```
// Example of using idatas.pos
// According the canvas position it will display a kind of grid.
float4 main( idatas i )
{
    float2 v = 60* frac( i.pos / 60 );
    float cx = v.x > 20 ? 1 : 0 ;
    float cy = v.y > 10 ? 1 : 0 ;
    float3 col = lerp( float3(0.92,0.25,0.07), float3(0.18,0.014,0.013), cx*cy ) ;
    return float4( col, 1 );
}
```



Potete trovare le spiegazioni di tutti i parametri (insieme a numerosi esempi) nella documentazione ufficiale che è consultabile tramite il menu **Help->Documentation** del **Brush Pixel Shader Editor**.

Esempi di BSL

Vi consiglio di provare i seguenti esempi e modificare i parametri e i numeri per vedere quali sono gli effetti sul pennello.

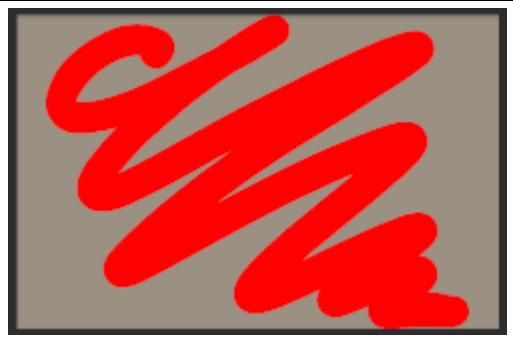
Errori nel codice BSL Quando modificate o create un BSL incontrerete molte segnalazioni di errore che via aiutano a scrivere il codice in maniera sintatticamente corretta. Finchè ci sono errori l'Anteprima del Pennello non mostra i risultati del nostro BSL.	<pre>File Edit Help 1 // Example of using canvasSize 2 float4 main(idatas i) 3 { 4 float2 p = saturate(i.pos / canvasSize) 5 return float4(p.x, p.y, 0, 1) ; 6 } 7 Syntax Error ';' expected</pre>
---	---

canvasSize <pre>// Example of using canvasSize float4 main(idatas i) { float2 p = saturate(i.pos / canvasSize); return float4(p.x, p.y, 0, 1) ; }</pre>	Anteprima del pennello mostrante una trama colorata.
---	--

```

viewZoom
// Example of using viewZoom
// the brush color will change according the zoom
value
float4 main( idatas i )
{
    float t = saturate( viewZoom );
    float3 col = lerp( float3(0,0,1),
float3(1,0,0), t );
    return float4( col, 1 );
}

```



```

viewBox
cfg{name="viewBox";}
float4 main( idatas i )
{
    float2 p = viewBox.toUpperLeftNorm( i.pos );
    float d = viewBox.getDistance( i.pos );
    d /= vmax( viewBox.size );
    d *= 50 ;
    float4 period = frac( d ) > 0.2 ?
float4(0.9,0.61,0.61,1) :
float4(0.92,0.32,0.19,1);
    period.xyz *= 1-exp( -abs(d)*0.5 );
    period = d < 0 ? period : period.yzxw;
    return float4( p.x, p.y, 1, 1 ) * period;
}

```



```

idatas.time
// Example of using idatas.time
float4 main( idatas i )
{
    float t = i.time;
    float4 col = i.color;
    col.xyz = 0.5 + 0.5*cos(t*3);
    return col ;
}

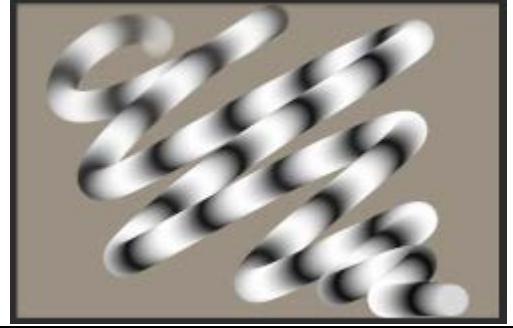
```



```

idata.dist
// Example of using idatas.dist
float4 main( idatas i )
{
    float t = i.dist/20;
    float4 col = i.color;
    col.xyz = 0.5 + 0.5*cos(t*3);
    return col;
}

```



```

perPrim-float2 (UI)
// According to a direction defined in a perPrim
parameter a color will display.
perPrim
{
    float2 dir = float2(1,0);
    {
        uiEditor = angleDist;
    }
}

float4 main( idatas i )

```



```

{
    float4 col1 = float4(0.9,0.7,0.3,1);
    float4 col2 = float4(0.1,0.3,0.97,1);
    float p = abs( dot( float2(0,1),
normalizeSafe(i.dir) ) );
    return lerp( col1, col2, p );
}

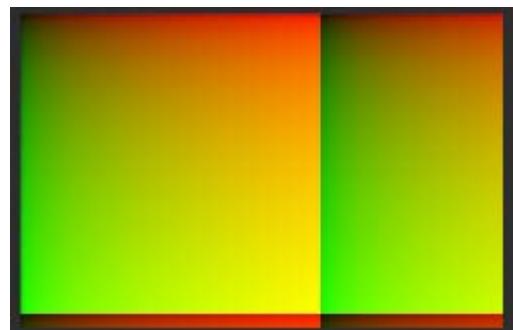
```



```

matrix2
// In this simple example the actual position will
be scaled by a factor 2, which will loop the
rendering every 150 px instead of 300 px
float4 main( idatas i )
{
    matrix2 test = matrix2FromScale( 2 );
    float2 p = test.transform( i.pos );
    p = max( (p % 300) / 300, 0 );
    return float4( p.x, p.y, 0, 1 );
}

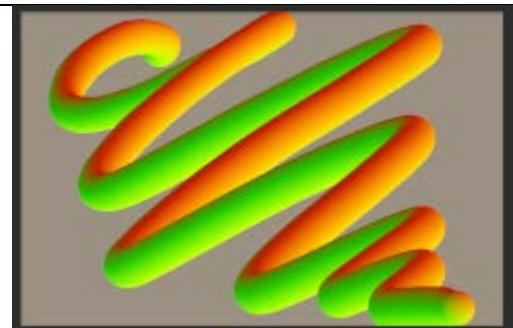
```



```

box2-toUpperLeftNorm
float4 main( idatas i )
{
    float2 p = i.primBox.toUpperLeftNorm( i.pos );
    return float4( p.x, p.y, 0, 1 );
}

```

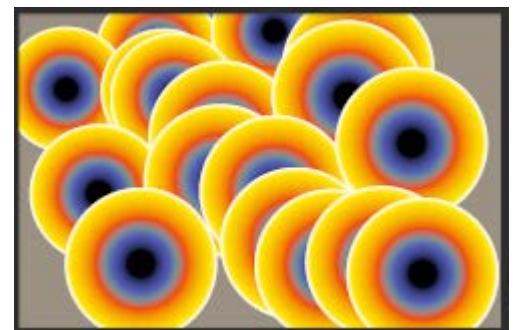


```

colorGradient
globals{
    colorGradient grad ;
}

float4 main( idatas i )
{
    float d = length( i.strokePos - i.pos ) / vmax(
0.5*i.primBox.size );
    float4 col = grad.sample( d );
    return col;
}

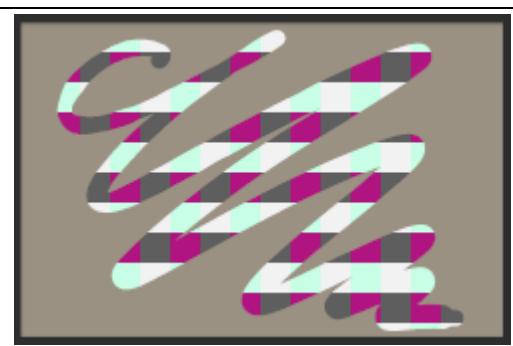
```



```

fromsRGB8
float4 main( idatas i )
{
    float2 p = frac( i.pos / 30 );
    float4 col = fromsRGB8(175,20,128);
    if( p.x < 0.5 )
        col.xyz = fromsRGB8( 94 );
    if( p.y < 0.5 )
        col.xyz = 1-col.xyz;
    return col ;
}

```



```

globals-float2-ui
globals{

    float2 test1 = float2(10.9,60.6);
{
    uiEditor = numeric;
    uiFormat = float3_2;
}

    float2 test2 = float2(101.9,900.6);
{
    uiEditor = angleDist;
}

    float2 test3 = float2(300.9,560.6);
{
    uiEditor = graph;
    uiYAxisInvert = true;
}
}

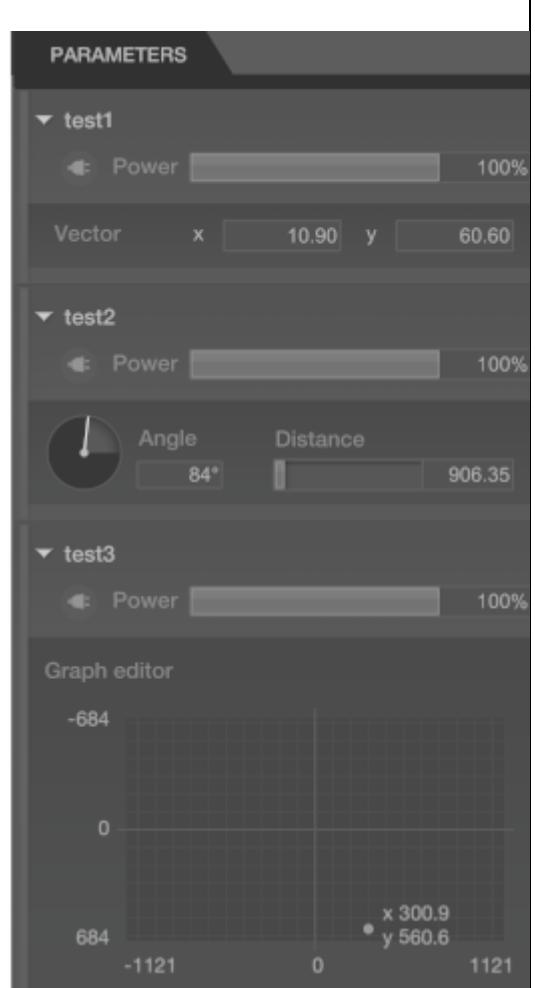
float4 main( idatas i )
{
    float d1 = distance( i.pos, test1 );
    float d2 = distance( i.pos, test2 );
    float d3 = distance( i.pos, test3 );

    float d = d1;
    float4 col = float4(1,0,1,1);
    if( d2 < d )
    {
        d = d2;
        col = float4( 0,1,1,1 );
    }
    if( d3 < d )
    {
        d = d3;
        col = float4( 0,0,1,1 );
    }

    col.xyz *= exp( -d*0.001 ) * lerp( 0.4, 1,
frac(d*0.01) );

    return col;
}

```



```

globals-float4-ui
globals{

    float4 col1 = float4(0.9,0.6,0.2,1);
    {
        uiEditor = numeric;
    }

    float4 col2 = float4(0.2,0.2,0.8,1);
    {
        uiEditor = color;
        uiColor = solid;
    }

    float4 col3 = float4(0.2,0.2,0.8,1);
    {
        uiEditor = color;
        uiColor = imageBased;
    }

    float4 col4 = float4(0.2,0.2,0.8,1);
    {
        uiEditor = color;
        uiColor = autoPicking;
    }

    float4 col5 = float4(0.2,0.2,0.8,1);
    {
        uiEditor = color;
        uiColor = gradient;
    }

    float4 col6 = float4(0.2,0.2,0.8,1);
    {
        uiEditor = color;
        uiColor = raw;
    }

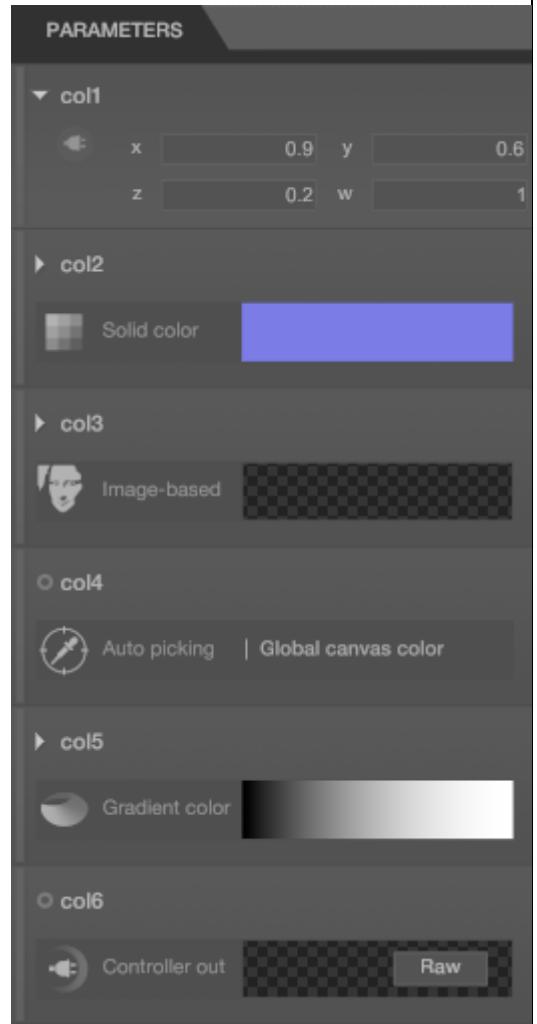
}

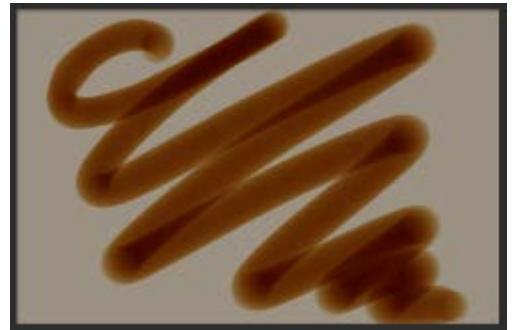
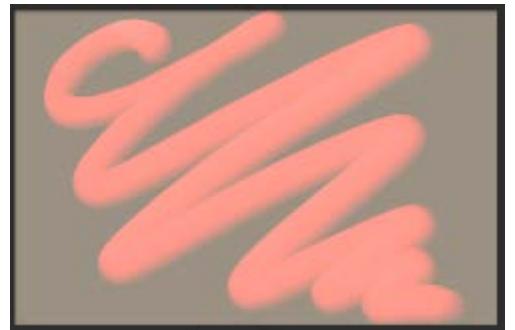
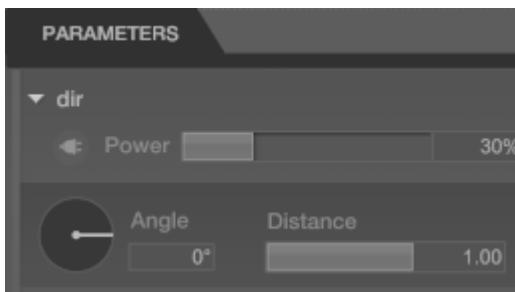
float4 main( idatas i )
{
    float2 p = frac( i.pos / 200.f ) ;

    float4 col;

    if( p.y > 0.5 )
    {
        if( p.x < 1./3. )
            col = col1 ;
        else if( p.x < 2./3. )
            col = col2;
        else if( p.x < 3./3. )
            col = col3;
    }
    else
    {
        if( p.x < 1./3. )

```



<pre> col = col4 ; else if(p.x < 2./3.) col = col5; else if(p.x < 3./3.) col = col6; } return col; } </pre>	
<pre> bottomLayer // makes a small effect of multiplying blending cfg{ blendEx = true; } float4 main(idatas i) { float4 col = float4(0.95, 0.88, 0.75, 1) * bottomLayer.pointSample(i, i.pos); return col ; } </pre>	
<pre> bottomLayer.pointSample // create a pixel shift effect with the addition of a red hue cfg{ blendEx = true; blendDefault = replace; samplingLayerMaxOffset = 1; } float4 main(idatas i) { float4 col = float4(0.09, 0.0045, 0.004, 1) + bottomLayer.pointSample(i, i.pos+1); return col; } </pre>	
<pre> onePixelColorSmudge cfg{ blendEx = true; blendDefault = replace; samplingLayerMaxOffset = 1; } globals{ float colorDensity = 2; { uiTab = "color"; // a default tab will be automatically created uiName = "Intensity"; uiEditor = slider; uiMin = 0; uiMax = 10; uiNumPixScale = 0.001; uiFormat = float3_2; } } perPrim{ // softness parameter } </pre>	<p>PARAMETERS</p>  

```

float hardness = 1;
{
    uiTab = "Shape";
    id = -1;
    uiMax = 1;
    uiName = "Hardness";
}

float2 dir = float2(1,0);
{
    uiEditor = angleDist;
}
}

float hardnessCpt( idatas i, float h )
{
    // get the min size of one of the axes of the
    primitives
    float sm = vmin( i.primBox.getSize() );
    // a negative distance means that we are inside
    it
    float d = max( -i.primDistance, 0 );
    // give a more linear feedback on this parameter
    float softness = pow( 1 - h,2 );
    // normalize the distance in function of the
    size of the primitive
    d /= sm*softness + 0.001;
    // attenuation curve function of distance
    return 1 - exp( -d );
}

float4 main( idatas i )
{
    float alpha = i.color.a*hardnessCpt( i,
    i.hardness );
    float4 coladd = i.color;
    coladd.xyz *= colorDensity * 0.01;
    float4 col = coladd+ bottomLayer.pointSample( i,
    i.pos+normalizeSafe(i.dir) );
    col.a = alpha*i.color.a;
    col = blendNormal( bottomLayer.pointSample( i,
    i.pos ), col );
    return col;
}

canvasNoise
cfg{
    renderingTime = 10;
}

perPrim{
    // parametre de densité du noise
    float density = 0.5;
    {
        uiName = "Density";
        uiMax = 1;
        uiTab = "Noise";
    }

    // parametre de hardness du noise
}

```



```

float densityHardness = 1;
{
    uiName = "Hardness";
    uiMax = 1;
    uiTab = "Noise";
}
// parametre de scale du noise
float noiseScale = 1;
{
    uiName = "Scale";
    uiMin = 0.25;
    uiMax = 100;
    uiTab = "Noise";
}
// parametre de softness
float alphaHardness = 1;
{
    uiName = "Hardness";
    uiMax = 1;
    uiTab = "Shape";
}
}

float hash( float3 n )
{
    return frac( sin(dot(n, float3(12.9898, 78.233,
36.791)))* 43758.5453 );
}

float adjustFromDensity( float r, float d )
{
    d = 1-d;
    float a = clamp( (d-0.5)*2, 0, 1 );
    float b = clamp( d*2, 0, 1 );
    r = (r-a) / max( b-a, 0.001 );
    r = clamp( r, 0, 1 );
    return r;
}

float hashD( float3 n, float d )
{
    return adjustFromDensity( hash( n ), d );
}

float noiseD( float3 x, float d )
{
    float3 f;
    float3 p = decompose( x, f );
    f = f*f*(3.0-2.0*f);
    float v1 = bilinearLerp(
hashD(p+float3(0,0,0),d),
hashD(p+float3(1,0,0),d),
            hashD(p+float3(0,1,0),d),
hashD(p+float3(1,1,0),d), f.xy );

    float v2 = bilinearLerp(
hashD(p+float3(0,0,1),d),
hashD(p+float3(1,0,1),d),

```



```

hashD(p+float3(0,1,1),d),
hashD(p+float3(1,1,1),d), f.xy );

    return lerp( v1, v2, f.z );
}

float hardnessCpt( idatas i, float h )
{
    // récupère la taille mini d'un des axes de la
    primitives
    float sm = vmin( i.primBox.getSize() )
    // une distance négative veut dire que l'on se
    trouve a l'intérieur de celle-ci;
    float d = max( -i.primDistance, 0 );
    // permet de donner un feedback plus linéaire
    sur ce paramètre
    float softness = pow( 1 - h
    // normalise la distance en fonction de la taille de
    la primitive,2);
    d /= sm*softness + 0.001;
    // courbe d'atténuation en fonction de la distance
    d = 1 - exp( -d );

    return i.primDistanceValid ? d : 1;
}

float4 main( idatas i )
{
    float density = i.density * hardnessCpt( i,
    i.densityHardness );
    float3 np = float3( toDrawSpace( i.pos ),
    toDrawSpace(i.dist) );
    np *= 1/i.noiseScale;
    float r = 0.5*noiseD( np, density ) ;
    r += 0.25*noiseD( np*2, density );
    r += 0.125*noiseD( np*4, density );
    r += 0.0625*noiseD( np*8, density );
    float4 color = i.color;
    color.a *= r * hardnessCpt( i, i.alphaHardness
);

    return color;
}

```

canvasSimpleVoronoi

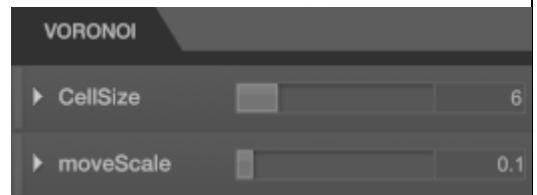
```

cfg{
    renderingTime = 10;
}

perPrim{
    float vSize = 20;
    {
        uiMin = 0.5;
        uiMax = 400;
        uiTab = "Voronoi";
        uiName = "CellSize";
    }

    float dScale = 0.1;
    {

```



```

uiMax = 1;
uiTab = "Voronoi";
uiName = "moveScale";
}

// parametre de softness
float alphaHardness = 1;
{
    uiName = "Hardness";
    uiMax = 1;
    uiTab = "Shape";
}
}

float hash( float2 n )
{
    float p = dot(n,float2(12.19898, 78.8233));
    return frac( sin( p )* 43758.5353 );
}

float2 hash2( float2 n )
{
    float2 p = float2( dot(n,float2(12.9898,
78.233)), dot(n,float2(26.95,18.33)) );
    return frac( sin( p )* 43758.5453 );
}

float voronoi( float2 x, float time )
{
    float2 f;
    float2 n = decompose( x, f );
    //-----
    // first pass: regular voronoi
    //-----
    float md = 8.0;
    float mc = 0;
    for( int j=-1; j<=1; j++ )
    {
        for( int i=-1; i<=1; i++ )
        {
            float2 g = float2( i, j );
            float2 o = hash2( n + g );
            float col = hash( n+g );
            o = 0.5 + 0.5*sin( time + 6.2831*o );  //
animate the position
            float2 r = g + o - f;
            float d = dot(r,r);
            if( d<md )
            {
                md = d;
                mc = col;
            }
        }
    }
    return mc;
}

float hardnessCpt( idatas i, float h )
{

```



```

// récupère la taille mini d'un des axes de la
primitives
float sm = vmin( i.primBox.getSize() );
// une distance négative veut dire que l'on se
trouve à l'intérieur de celle-ci
float d = max( -i.primDistance, 0 );
// permet de donner un feedback plus linéaire
sur ce paramètre
float softness = pow( 1 - h, 2 );
// normalise la distance en fonction de la taille de
la primitive
d /= sm*softness + 0.001;
// courbe d'atténuation en fonction de la distance
d = 1 - exp( -d );
return i.primDistanceValid ? d : 1;
}

float4 main( idatas i )
{
    float2 p = toDrawSpace( i.pos );
    p /= i.vSize;
    float c = voronoi( p, i.dScale*0.1*toDrawSpace(
i.dist ) );
    float4 col = i.color;
    // col.xyz = c;
    col.a *= c * hardnessCpt( i, i.alphaHardness );
    return col;
}

```

canvasEdgeVoronoi

```

cfg{
    renderingTime = 20;
}

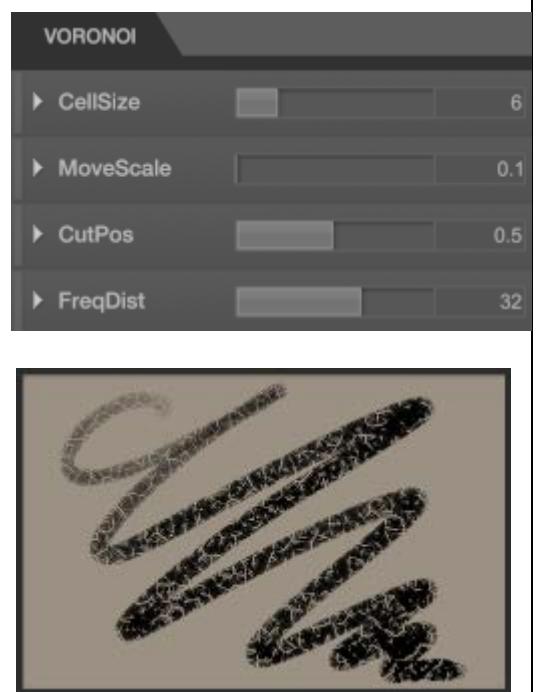
perPrim{
    float vSize = 20;
    {
        uiMin = 0.5;
        uiMax = 400;
        uiTab = "Voronoi";
        uiName = "CellSize";
    }

    float dScale = 0.01;
    {
        uiMin = 0.001;
        uiMax = 10;
        uiTab = "Voronoi";
        uiName = "MoveScale";
    }

    float cutPos = 0.5;
    {
        uiMax = 1;
        uiTab = "Voronoi";
        uiName = "CutPos";
    }

    float freqDist = 32;
    {

```



```

uiMax = 256;
uiTab = "Voronoi";
uiName = "FreqDist";
}

// parametre de softness
float alphaHardness = 1;
{
    uiName = "Hardness";
    uiMax = 1;
    uiTab = "Shape";
}
}

float2 hash2( float2 n )
{
    float2 p = float2( dot(n,float2(12.9898,
78.233)), dot(n,float2(26.95,18.33)) );
    return frac( sin( p )* 43758.5453 );
}

float3 voronoi( float2 x, float time )
{
    float2 f;
    float2 n = decompose( x, f );
    //-----
    // first pass: regular voronoi
    //-----
    float2 mg;
    float2 mr;

    float md = 8.0;
    for( int j=-1; j<=1; j++ )
    {
        for( int i=-1; i<=1; i++ )
        {
            float2 g = float2( i, j );
            float2 o = hash2( n + g );
            o = 0.5 + 0.5*sin( time + 6.2831*o ); // animate the position
            float2 r = g + o - f;
            float d = dot(r,r);

            if( d<md )
            {
                md = d;
                mr = r;
                mg = g;
            }
        }
    }

    //-----
    // second pass: distance to borders
    //-----
    md = 8.0;
    float2 mr2;
    for( int j=-2; j<=2; j++ )

```



```

{
    for( int i=-2; i<=2; i++ )
    {
        float2 g = float2( i, j );
        float2 o = hash2( n + g );
        o = 0.5 + 0.5*sin( time + 6.2831*o ); // animate the position
        float2 r = g + o - f;

        if( dot(mr-r,mr-r)>0.00001 )
        {
            float dist = dot( 0.5*(mr+r), normalize(r-mr) );
            if( dist < md )
            {
                md = dist;
                mr2 = r;
            }
        }
    }
    return float3( md, -normalize(mr-mr2) );
}

float hardnessCpt( idatas i, float h )
{
    // récupère la taille mini d'un des axes de la primitive
    float sm = vmin( i.primBox.getSize() );
    // une distance négative veut dire que l'on se trouve à l'intérieur de celle-ci
    float d = max( -i.primDistance, 0 );
    // permet de donner un feedback plus linéaire sur ce paramètre
    float softness = pow( 1 - h,2 );
    // normalise la distance en fonction de la taille de la primitive
    d /= sm*softness + 0.001;
    d = 1 - exp( -d ); // courbe d'atténuation en fonction de la distance
    return i.primDistanceValid ? d : 1;
}

float4 main( idatas i )
{
    float2 p = toDrawSpace( i.pos );
    p /= i.vSize;
    float3 c = voronoi( p, i.dScale*toDrawSpace( i.dist ) );
    float4 col = i.color;
    col.a *= hardnessCpt( i, i.alphaHardness );
    col.a *= (0.5 + 0.5*sin( c.x*i.freqDist ) ) > i.cutPos ? 0 : 1;
    return col;
}

```

```

canvasTextureVoronoiTiling
globals{
    texture mat;
}

float hash1( float2 p)
{
    float hscale = .1031;
    float3 p3 = frac( float3(p.x,p.y,p.x) * hscale
);
    p3 += dot(p3, p3.yzx + 19.19);
    return frac( (p3.x + p3.y) * p3.z );
}

float2 hash2( float2 p )
{
    float3 hscale3 = float3( .1031, .1030, .0973 );
    float3 p3 = frac( float3(p.x,p.y,p.x) *
hscale3);
    p3 += dot(p3, p3.yzx+19.19);
    return frac( float2( (p3.x + p3.y)*p3.z,
(p3.x+p3.z)*p3.y ) );
}

float4 hash4( float2 p )
{
    float4 hscale4 = float4( .1031, .1030, .0973,
.1099 );
    float4 p4 = frac( float4(p.xyxy) * hscale4 );
    p4 += dot( p4, p4.wzxy+19.19);
    return frac( float4((p4.x + p4.y)*p4.z, (p4.x +
p4.z)*p4.y, (p4.y + p4.z)*p4.w, (p4.z +
p4.w)*p4.x));
}

float4 voronoiTex( texture tex, float2 uv )
{
    float2 f;
    float2 n = decompose( uv, f );
    // derivatives ( for correct mipmapping )
    float2 Dx = ddx(uv);
    float2 Dy = ddy(uv);
    -----
    // regular voronoi
    -----
    float4 tot = 0;
    float totAlpha = 0;
    for( int j=-1; j<=1; j++ )
    {
        for( int i=-1; i<=1; i++ )
        {
            float2 g = float2( i, j );
            float2 ipos = n + g;
            float2 o = hash2( ipos );
            float2 r = g + o - f;
            float d = dot(r,r);
            float alpha = exp(-15*d);
            // Génère les paramètres de transformation
de la texture
}
}

```



```

        float4 trans = hash4( ipos );
        float2 dir = sign( trans.xy-0.5 );
        float4 col = tex.sampleGrad( uv*dir +
trans.zw, Dx, Dy );
        tot += alpha*col;
        totAlpha += alpha;
    }
}
return tot / totAlpha;
}

float4 main( idatas i )
{
    float4 col = voronoiTex( mat, 10*i.pos/mat.size
);
    col = mat.isEmpty ? 1 : col;
    return col;
}

```

canvasRoughness

```

globals{
    uiTab stroke;
    { row = 99; }

    uiTab roughness;
    { row = 100; }

    texture mat;
    {
        uiTab = "roughness";
        uiName = "map";
    }

    float shadowStrength = 0.5;
    {
        uiMin = 0;
        uiMax = 1;
        uiName = "SShadow strenght";
        uiTab = "roughness";
        id = 100;
    }
}

perPrim {
    float pushPressure = 0.5;
    {
        uiMin = 0;
        uiMax = 1;
        uiFormat = percent;
        uiName = "pressure";
        uiTab = "stroke";
    }

    // direction du tracé actuel
    float2 dir;
    {
        uiEditor = angleDist;
        raw = true;
        uiTab = "stroke";
    }
}
```



```

float canvasHeight = 0.5;
{
    uiMin = 0;
    uiMax = 1;
    uiFormat = percent;
    uiName = "roughness";
    uiTab = "roughness";
}

float shadowSize = 5;
{
    uiMin = 0;
    uiMax = 40;
    uiName = "SShadow size";
    uiTab = "roughness";
    id = 101;
}

float rPower = 0.85;
{
    uiMin = 0;
    uiMax = 1;
    uiFormat = percent;
    uiName = "splat strenght";
    uiTab = "roughness";
}

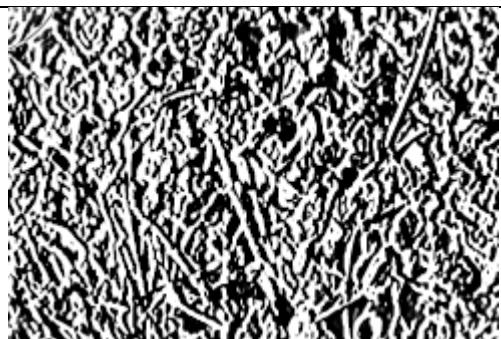
float hardness = 1;
{
    uiTab = "Shape";
    id = -1;
    uiMax = 1;
    uiName = "Hardness";
}
}

float surfaceHeight( float2 uv )
{
    return luminance( mat.sample( uv ) );
}

// with derivative
float surfaceHeight( float2 uv, float2 dx, float2
dy )
{
    return luminance( mat.sampleGrad( uv, dx, dy ) );
}

float3 surfaceNormal( float2 pos, float h )
{
    float2 opa = 1/mat.size;
    float2 uv = pos / mat.size;
    // compute normal with a Sobel filter
    float s00 = surfaceHeight( uv + opa*float2(-1,-
1) );
    float s10 = surfaceHeight( uv + opa*float2( 0,-
1) );
}

```



```

    float s20 = surfaceHeight( uv + opa*float2( 1,-
1) );
    float s01 = surfaceHeight( uv + opa*float2(-1,
0) );
    float s21 = surfaceHeight( uv + opa*float2( 1,
0) );
    float s02 = surfaceHeight( uv + opa*float2(-1,
1) );
    float s12 = surfaceHeight( uv + opa*float2( 0,
1) );
    float s22 = surfaceHeight( uv + opa*float2( 1,
1) );
    // Compute dx using Sobel:
    //      -1 0 1
    //      -2 0 2
    //      -1 0 1
    float dX = -s00 + s20 -2*s01 + 2*s21 -s02 + s22;
    // Compute dy using Sobel:
    //      -1 -2 -1
    //      0  0  0
    //      1  2  1
    float dY = -s00 -2*s10 -s20 + s02 + 2*s12 + s22;
    return normalizeSafe( float3( h*dX, h*dY, 1 ) );
}

float Shadow( float2 pos, float2 dir, float
shadowSize, float h )
{
    float2 uv = pos / mat.size;
    float2 uvStart = (pos - dir*shadowSize ) /
mat.size;
    int nbstep = 6;
    float2 dx = ddx( uv );
    float2 dy = ddy( uv );
    float3 start3 = float3( uvStart, 1 );
    float3 end3 = float3( uv, h*surfaceHeight( uv )
);
    float opacity = 1;
    for( int a=0; a<nbstep; a++ )
    {
        float u = (a+0.5)/nbstep;
        float dxyscale = lerp( 1+shadowSize*2, 1, u );
        float3 cur = lerp( start3, end3, u );
        float ch = h*surfaceHeight( cur.xy,
dx*dxyscale, dy*dyyscale ) - 0.01;
        opacity *= 1-shadowStrength*exp( -5*saturate(
cur.z - ch ) );
    }
    return opacity;
}

float hardnessCpt( idatas i, float h )
{
    float sm = vmin( i.primBox.getSize() );
    float d = max( -i.primDistance, 0 );
    float softness = pow( 1 - h,2 );
    d /= sm*softness + 0.001;
    d = 1 - exp( -d );
    d = i.primDistanceValid ? d : 1;
}

```

```

    return d;
}

float4 main( idatas i )
{
    float heightscale = 8;
    float2 dir = normalizeSafe( i.dir );
    float3 bn = normalizeSafe(
float3(heightscale*(1-i.pushPressure)*dir,-1) );
    float3 cn = surfaceNormal( i.pos,
i.canvasHeight*heightscale );
    float a = saturate( -dot( bn, cn ) );
    a = pow( a, lerp( 0.1, 8, 1-i.rPower ) );
    a *= Shadow( i.pos, dir, i.shadowSize,
i.canvasHeight );
    a*= hardnessCpt( i, i.hardness );
    float4 col = i.color;
    col.a *= a;
    return col;
}

```

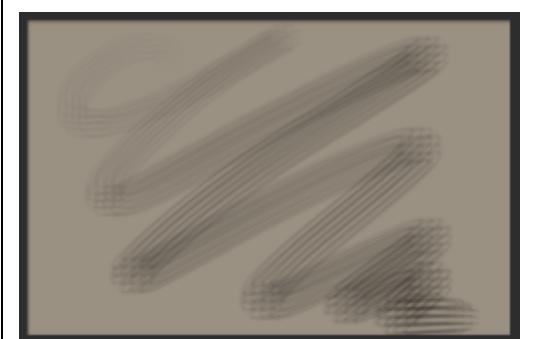
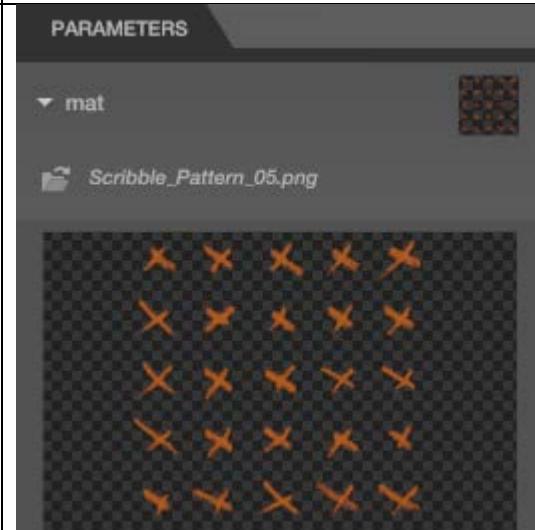
primTextureAlphaLuminance

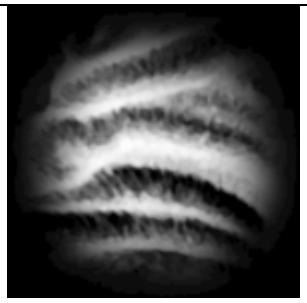
```

globals{
    texture mat;
}

float4 main( idatas i )
{
    float2 p = i.primBox.toUpperLeftNorm( i.pos );
    float4 cmat = mat.sample( p );
    cmat = mat.isEmpty ? 1 : cmat;
    float4 col = i.color;
    col.a *= luminance( cmat );
    return col;
}

```





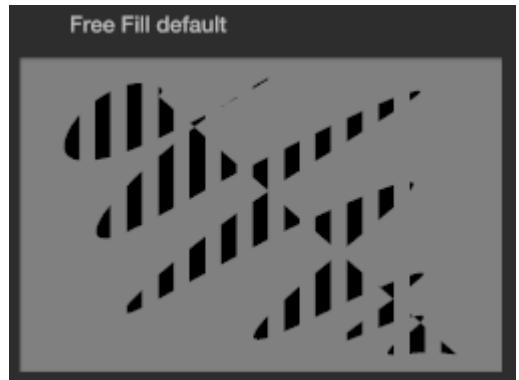
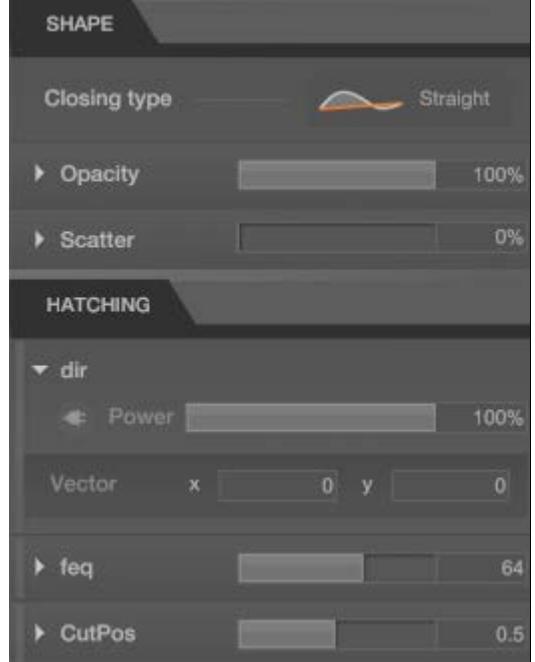
hatchingDir

```
perPrim{
    float2 dir;
{
    uiTab = "Hatching";
}

float freq = 64;
{
    uiMax = 1000;
    uiTab = "Hatching";
    uiName = "freq";
}

float cutPos = 0.5;
{
    uiMax = 1;
    uiTab = "Hatching";
    uiName = "CutPos";
}
}

float4 main( idatas i )
{
    plane2 p = plane2FromNormPos(
normalizeSafe(i.dir), i.strokePos );
    float d = toDrawSpace( p.getDistance( i.pos ) );
    float4 col = i.color;
    col.a *= frac( d*i.freq/1000 ) > i.cutPos ? 1 :
0;
    //col.xy = normalizeSafe(i.dir);
    return col;
}
```



```

attenuationDir
perPrim{

    float2 dir;
    {
        uiTab = "Gradient";
        uiName = "Dir";
    }

    float mydist = 128;
    {
        uiMin = 0.01;
        uiMax = 1000;
        uiTab = "Gradient";
        uiName = "mydist";
    }

    float4 main( idatas i )
    {
        plane2 p = plane2FromNormPos(
normalizeSafe(i.dir), i.strokePos );
        float d = toDrawSpace( p.getDistance( i.pos ) );
        float4 col = i.color;
        col.a *= 1-clamp( abs(d)/i.mydist, 0, 1 );
        return col;
    }
}

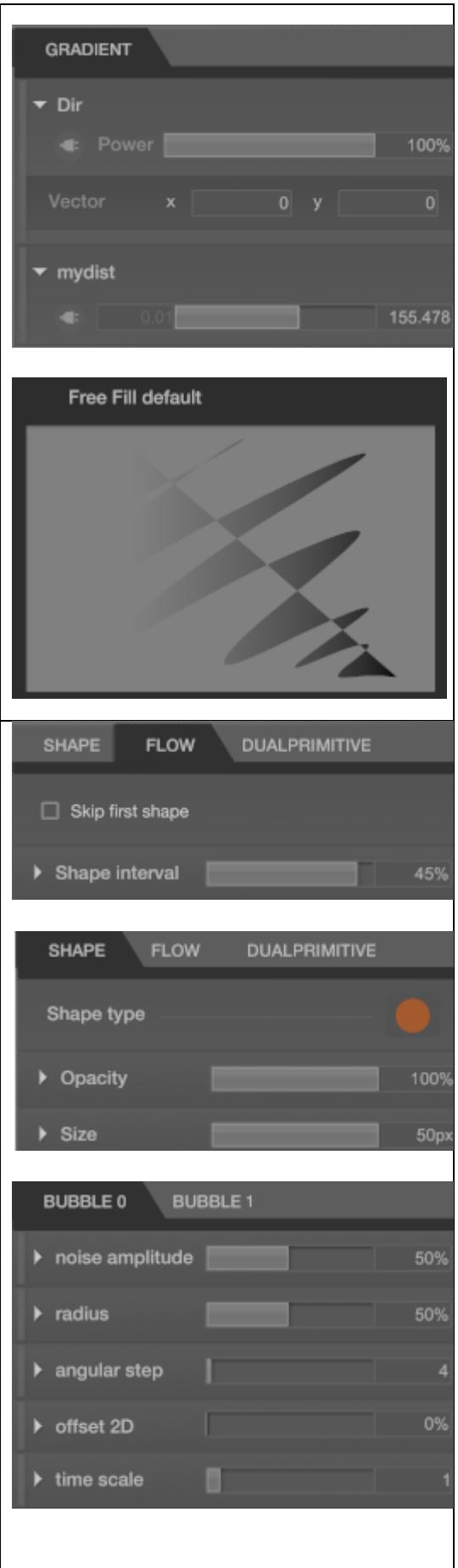
bubbleNoise
globals{
    float fnbStep0 = 4;
    {
        uiMin = 3;
        uiMax = 64;
        uiTab = "bubble 0";
        uiFormat = integer;
        uiName = "angular step";
        id = 2;
    }

    float fnbStep1 = 8;
    {
        uiMin = 3;
        uiMax = 64;
        uiTab = "bubble 1";
        uiFormat = integer;
        uiName = "angular step";
        id = 2;
    }

    float tscale0 = 1;
    {
        uiMin = 0;
        uiMax = 10;
        uiTab = "bubble 0";
        uiName = "time scale";
        id = 120;
    }

    float tscale1 = 1;
}

```



```

{
    uiMin = 0;
    uiMax = 10;
    uiTab = "bubble 1";
    uiName = "time scale";
    id = 120;
}
}

perPrim {
    float ampl0 = 0.5;
{
    uiMin = 0;
    uiMax = 1;
    uiTab = "bubble 0";
    uiFormat = percent;
    uiName = "noise amplitude";
    id = 0;
}

float ampl1 = 0.25;
{
    uiMin = 0;
    uiMax = 1;
    uiTab = "bubble 1";
    uiFormat = percent;
    uiName = "noise amplitude";
    id = 0;
}

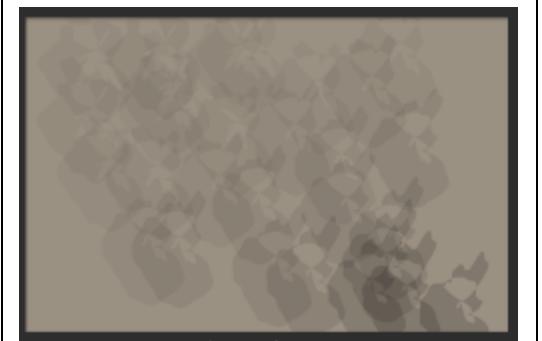
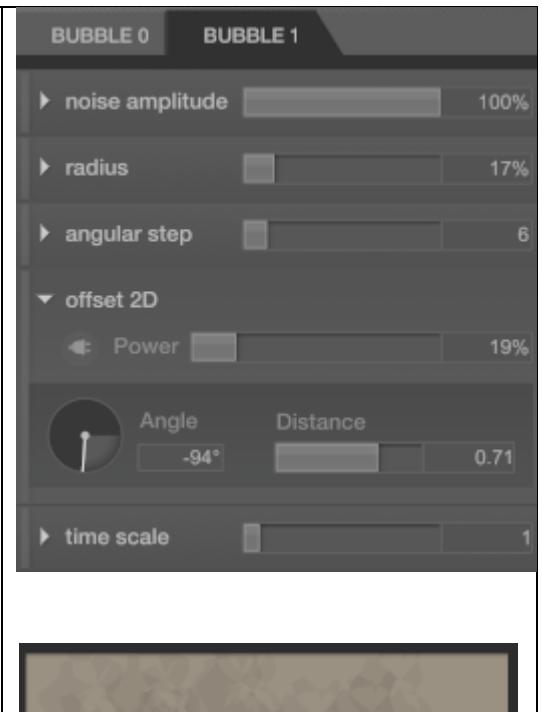
float aoff0 = 0.5;
{
    uiMin = 0;
    uiMax = 1;
    uiTab = "bubble 0";
    uiFormat = percent;
    uiName = "radius";
    id = 1;
}

float aoff1 = 0.25;
{
    uiMin = 0;
    uiMax = 1;
    uiTab = "bubble 1";
    uiFormat = percent;
    uiName = "radius";
    id = 1;
}

float2 coff0 = 0.5;
{
    uiTab = "bubble 0";
    uiName = "offset 2D";
    uiEditor = angleDist;
    power = 0;
}

float2 coff1 = -0.5;

```



```

{
    uiTab = "bubble 1";
    uiName = "offset 2D";
    uiEditor = angleDist;
    power = 0;
}
}

float hash1( float2 p)
{
    float hscale = .1031;
    float3 p3 = frac( float3(p.x,p.y,p.x) * hscale
);
    p3 += dot(p3, p3.yzx + 19.19);
    return frac( (p3.x + p3.y) * p3.z );
}

float noise( float2 p, float t )
{
    float f;
    float ip = decompose( p.x, f );
    float ip1 = mod( ip+1, p.y ); // fait boucler
en fct du maxi passé en y
    float ft;
    float it = decompose( t, ft );
    float it1 = it+1;
    float a0 = smoothLerp( hash1(float2(ip,it)),
hash1(float2(ip1,it)), f );
    float a1 = smoothLerp( hash1(float2(ip,it1)),
hash1(float2(ip1,it1)), f );
    return smoothLerp( a0, a1, ft );
}

// return 1 if we are in the AngleCurve shape
float AngleCurveShape( float2 pos, float2 offset,
float ampl, float aoffset, float time, float
fnbStep, float seed )
{
    float2 sph = toSpherical( pos-offset );
    float angle = mod( sph.x + seed , TWOPi );
    float nbStep = floor(fnbStep);
    angle = nbStep * angle / TWOPi;
    float displ = 0.5*noise( float2(angle,
nbStep), time )
        + 0.25*noise( 2*float2(angle, nbStep),
time )
        + 0.125*noise( 4*float2(angle, nbStep),
time )
        + 0.0625*noise( 8*float2(angle, nbStep),
time );
    float p = aoffset + ampl*(2*displ-1);
    return sph.y > p ? 0 : 1;
}

float4 main( idatas i )
{
    float2 uv = i.primBox.toUpperLeftNorm( i.pos
)*2-1;
//    float2 sph = toSpherical( uv );
}

```

```

    float seed = i.nbUserStroke * 2.23881;
    float shape0 = AngleCurveShape( uv, i.coff0,
i.ampl0, i.aoff0, i.time*tscale0 + seed, fnbStep0,
seed );
    float shape1 = AngleCurveShape( uv, i.coff1,
i.ampl1, i.aoff1, i.time*tscale1 + seed*0.63798,
fnbStep1, seed*1.205 );
    float gshape = shape0 + shape1;
    float alpha = gshape == 1 ? 1 : 0;
    float4 col = i.color;
    col.a *= alpha;
//  col.xyz = mod( sph.x, TWOPI ) / TWOPI;
//  col.xyz = iicut;
//  col.xyz = mod( i.nbUserStroke, 10 ) / 10;
    return col;
}

pointShadow
perPrim {

    float size = 0.5;
{
    uiMin = 0;
    uiMax = 1;
    uiTab = "shape";
    uiFormat = percent;
    uiName = "inner size";
}

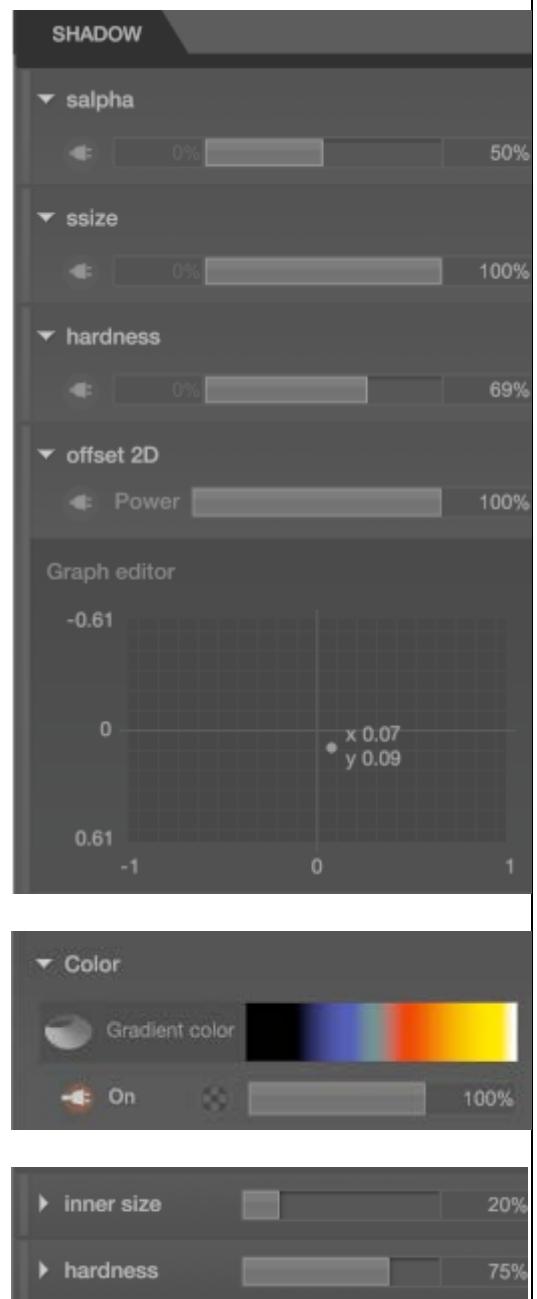
    float hardness = 0;
{
    uiMin = 0;
    uiMax = 1;
    uiTab = "shape";
    uiFormat = percent;
}

    float salpha = 0.5;
{
    uiMin = 0;
    uiMax = 1;
    uiTab = "shadow";
    uiFormat = percent;
}

    float ssize = 0.1;
{
    uiMin = 0;
    uiMax = 1;
    uiTab = "shadow";
    uiFormat = percent;
}

    float shardness = 0.5;
{
    uiMin = 0;
    uiMax = 1;
    uiTab = "shadow";
    uiFormat = percent;
    uiName = "hardness";
}

```



```

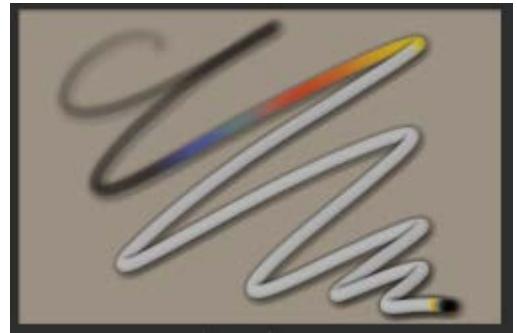
}

float2 soffset = 0.05;
{
    uiTab = "shadow";
    uiName = "offset 2D";
    uiEditor = graph;
    uiYAxisInvert = true;
    power = 1;
}
}

float cirlceGauss( float2 p, float r, float
hardness )
{
    float d = max( -(length( p ) - r), 0 );
//    return d;
    // permet de donner un feedback plus lineaire
    // sur ce paramètre
    float softness = pow( 1 - hardness,2 );
    // normalise la distance en fct de la taille de
    // la primitive
    d /= r*softness + 0.0001;
    // courbe d'atténuation en fct de la distance
    return 1 - exp( -d );
}

float4 main( idatas i )
{
    float2 uv = i.primBox.toUpperLeftNorm( i.pos
)*2-1;
    float shadow = i.salpha * i.color.a *
cirlceGauss( uv - i.soffset, i.size*(1+i.ssize),
i.hardness * i.shardness );
    float4 colShadow = float4(0,0,0,shadow );
    float calpha = cirlceGauss( uv, i.size,
i.hardness );
    float4 col = i.color;
    col *= calpha;
    col = blendNormal( colShadow, col );
    return col;
}

```



```

primSimpleNoise
cfg{
    renderingTime = 5;
}

globals{
    float noiseScale = 1;
    {
        uiTab = "Noise";
        uiName = "Scale";
        uiFormat = percent;
        uiMax = 1;
    }
}

perPrim{
    float ndensity = 0.5;
    {
        uiTab = "Noise";
        uiName = "Density";
        uiFormat = percent;
        uiMax = 1;
    }

    float densityHardness = 0.15;
    {
        uiName = "Hardness";
        uiMax = 1;
        uiTab = "Noise";
    }
}

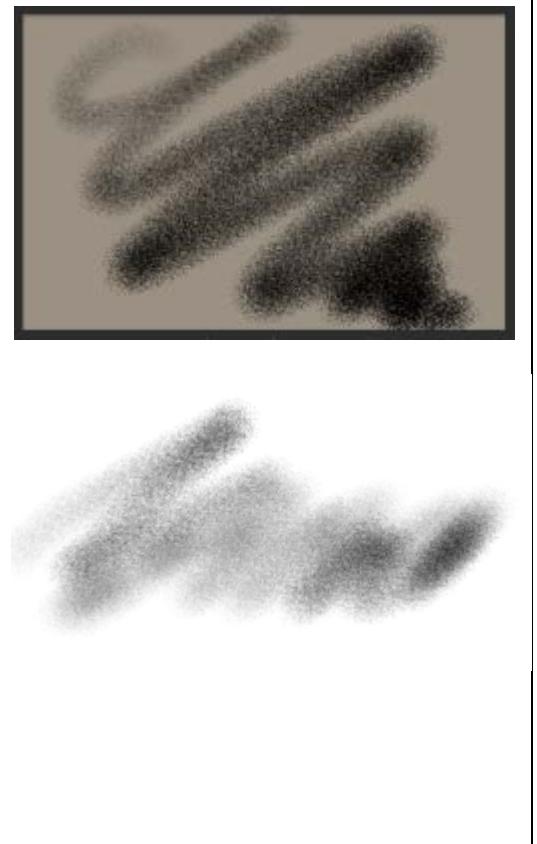
float hash( float3 p )
{
    float hscale = .1031;
    float3 p3 = frac( p * hscale );
    p3 += dot(p3, p3.yzx + 19.19);
    return frac( (p3.x + p3.y) * p3.z );
}

float adjustDensity( float r, float d )
{
    d = 1-d;
    float a = clamp( (d-0.5)*2, 0, 1 );
    float b = clamp( d*2, 0, 1 );
    r = (r-a) / max( b-a, 0.001 );
    r = saturate( r );
    return r;
}

float hashD( float3 p, float d )
{
    return adjustDensity( hash(p), d );
}

float hardnessCpt( idatas i, float h )
{
    float sm = vmin( i.primBox.getSize() );
    float d = max( -i.primDistance, 0 );
}

```



```

        float softness = pow( 1 - h,2 );
        d /= sm*softness + 0.001;
        d = 1 - exp( -d );
        return i.primDistanceValid ? d : 1;
    }

float cptPrimNoise( idatas i, float time, float
density )
{
    float2 p = i.primBox.toUpperLeftNorm( i.pos )*2
- 1;
    p = floor( p*i.primBox.getSize()*noiseScale );
    return hashD( float3( p, time ), density );
}

float4 main( idatas i )
{
    float density = i.ndensity * hardnessCpt( i,
i.densityHardness );
    float alpha = cptPrimNoise( i, i.dist*0.001,
density );
    float4 col = i.color;
    col.a *= alpha;
    return col;
}

primNoise
cfg{
    renderingTime = 30;
}

perPrim{

    // parametre de densité
    float density = 0.35;
    {
        uiName = "Density";
        uiMax = 1;
        uiTab = "Noise";
    }

    // parametre de softness du noise
    float densityHardness = 0.15;
    {
        uiName = "Hardness";
        uiMax = 1;
        uiTab = "Noise";
    }

    // parametre de scale
    float noiseScale = 2;
    {
        uiName = "Scale";
        uiMin = 0.25;
        uiMax = 100;
        uiTab = "Noise";
    }

    // parametre de noise qui se modifie plus ou
moins rapidement en fonction de la distance
}

```



```

float evoSpeed = 0.05;
{
    uiName = "EvoSpeed";
    uiMax = 1;
    uiTab = "Noise";
}

// parametre de softness
float alphaHardness = 1;
{
    uiName = "Hardness";
    uiMax = 1;
    uiTab = "Shape";
}
}

float hash( float3 n )
{
    return frac( sin(dot(n, float3(12.9898, 78.233,
36.791)))* 43758.5453 );
}

float adjustFromDensity( float r, float d )
{
    d = 1-d;
    float a = clamp( (d-0.5)*2, 0, 1 );
    float b = clamp( d*2, 0, 1 );
    r = (r-a) / max( b-a, 0.001 );
    r = clamp( r, 0, 1 );
    return r;
}

float hashD( float3 n, float d )
{
    return adjustFromDensity( hash( n ), d );
}

float noiseD( float3 x, float d )
{
    float3 f;
    float3 p = decompose( x, f );
    f = f*f*(3.0-2.0*f);
    float v1 = bilinearLerp(
hashD(p+float3(0,0,0),d),
hashD(p+float3(1,0,0),d),
            hashD(p+float3(0,1,0),d),
            hashD(p+float3(1,1,0),d), f.xy );
    float v2 = bilinearLerp(
hashD(p+float3(0,0,1),d),
hashD(p+float3(1,0,1),d),
            hashD(p+float3(0,1,1),d),
            hashD(p+float3(1,1,1),d), f.xy );
    return lerp( v1, v2, f.z );
}

float hardnessCpt( idatas i, float h )
{
    // récupère la taille mini d'un des axes de la
primitives
}

```



```

float sm = vmin( i.primBox.getSize() );
// une distance négative veut dire que l'on se
trouve a l'intérieur de celle-ci
float d = max( -i.primDistance, 0 );
// permet de donner un feedback plus linéaire
sur ce paramètre
float softness = pow( 1 - h,2 );
// normalise la distance en fonction de la taille de
la primitive
d /= sm*softness + 0.001;
// courbe d'atténuation en fonction de la distance
d = 1 - exp( -d );
return i.primDistanceValid ? d : 1;
}

float4 main( idatas i )
{
    // récupère la matrice de transformation qui va
    passer de l'espace
    // canvas à l'espace en haut à gauche de la box
    matrix2 t = i.primBox.getToUpperLeftTransfo();
    // transforme la position courante dans l'espace
    de la box
    float2 p = t.transform( i.pos );
    // normalise la position par rapport à la box
    p /= i.primBox.getSize();
    p *= i.noiseScale;
    float density = i.density * hardnessCpt( i,
    i.densityHardness );
    float3 np = float3( p,
    toDrawSpace(i.dist)*i.evoSpeed*0.01 );
    // np *= 1/i.noiseScale;
    float r = 0.5*noiseD( np, density );
    r += 0.25*noiseD( np*2, density );
    r += 0.125*noiseD( np*4, density );
    r += 0.0625*noiseD( np*8, density );
    float4 color = i.color;
    color.a *= r * hardnessCpt( i, i.alphaHardness
);
    return color;
}

```

blur

```

cfg{
    blendEx = true;
    blendDefault = replace;
    samplingLayerMaxOffset = 10;
    renderingTime = 50;
}

```

perPrim{

```

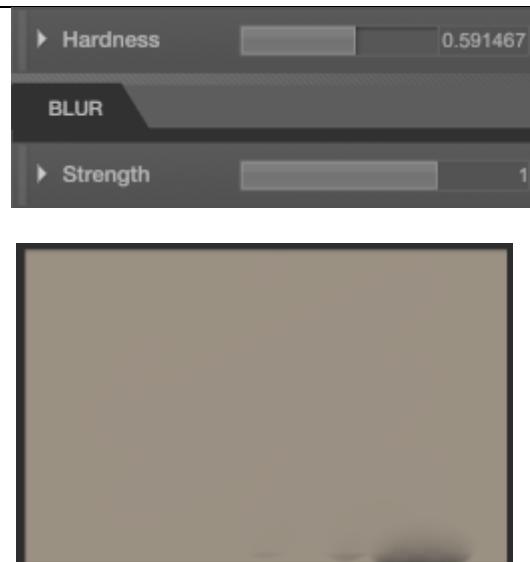
    float blurStrength = 1;
    {
        uiTab = "Blur";
        uiName = "Strength";
        uiMax = 1;
    }

```

```

    float alphaHardness = 1;

```



```

{
    uiName = "Hardness";
    uiMax = 1;
    uiTab = "Shape";
}

}

float hardnessCpt( idatas i, float h )
{
    // récupère la taille mini d'un des axes de la
    primitives
    float sm = vmin( i.primBox.getSize() );
    // une distance négative veut dire que l'on se
    trouve a l'intérieur de celle-ci;
    float d = max( -i.primDistance, 0 );
    // permet de donner un feedback plus linéaire
    sur ce paramètre
    float softness = pow( 1 - h,2 );
    // normalise la distance en fonction de la taille de
    la primitive,2);
    d /= sm*softness + 0.001;
    // courbe d'atténuation en fonction de la distance
    d = 1 - exp( -d );
    return i.primDistanceValid ? d : 1;
}

float4 main( idatas i )
{
    int nbsample = 7;
    float4 noBlur = bottomLayer.pointSample( i,
    i.pos );
    float maxdec = nbsample;
    float radius = i.blurStrength * fromDrawSpace(
    maxdec );
    radius = min( radius, maxdec );
    float tot = 0;
    float4 coltot = 0;
    for( int y=0; y<nbsample; y++ )
    {
        for( int x = 0; x<nbsample; x++ )
        {
            float2 d2 = 2*( ( float2( x, y ) + 0.5 ) /
    nbsample ) - 0.5 );
            float d = length( d2 );
            float att = exp( -d );
            tot += att;
            coltot += att * bottomLayer.bilinearSample(
    i, i.pos + d2*radius );
        }
    }
    coltot /= tot;
    float alpha = i.color.a * hardnessCpt( i,
    i.alphaHardness );
    alpha *= smoothLerp( 0, 1, min( radius, 1 ) );
    float4 col = blendNormalAlpha( noBlur, coltot,
    alpha );
    return col;
}

```



```

}

smudge
cfg{
    blendEx = true;
    blendDefault = replace;
    samplingLayerMaxOffset = 16;
    renderingTime = 30;
}

perPrim {

    float hardness = 0.66;
    {
        uiMin = 0;
        uiMax = 1;
        uiTab = "smudge";
        uiFormat = percent;
        uiName = "hardness";
    }

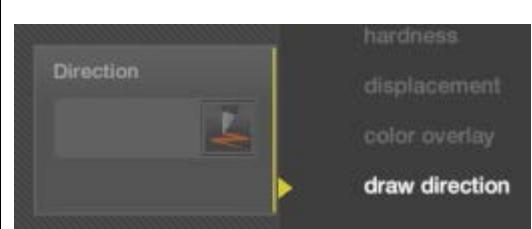
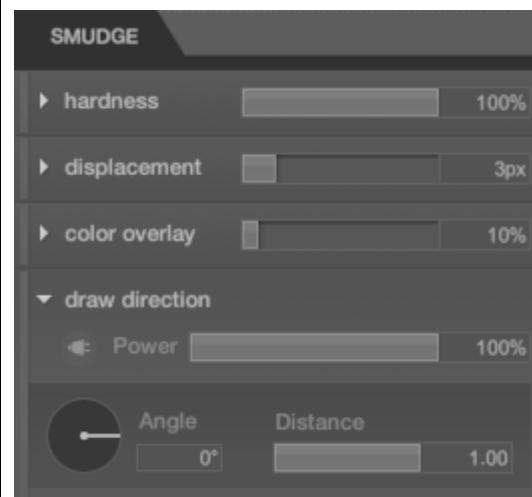
    float maxDisplace = 3;
    {
        uiMin = 0;
        uiMax = 16;
        uiTab = "smudge";
        uiFormat = pixel;
        uiName = "displacement";
    }

    float overlayFactor = 0.1;
    {
        uiMax = 1;
        uiTab = "smudge";
        uiFormat = percent;
        uiName = "color overlay";
    }

    float2 dir = float2(1,0);
    {
        uiTab = "smudge";
        uiName = "draw direction";
        uiEditor = angleDist;
    }
}

float smudgeFactor( idatas i )
{
    // min axe size of the primitive
    float sm = vmin( i.primBox.getSize() );
    // distance to the primitive;
    float d = max( -i.primDistance, 0 );
    d = i.primDistanceValid ? d : 0;
    // more linear feedback from the parameter
    float softness = pow( 1 - i.hardness,2 );
    // normalize distance with the primitive size
    d /= sm*softness + 0.0001;
    // attenuation curve
    return 1 - exp( -d );
}

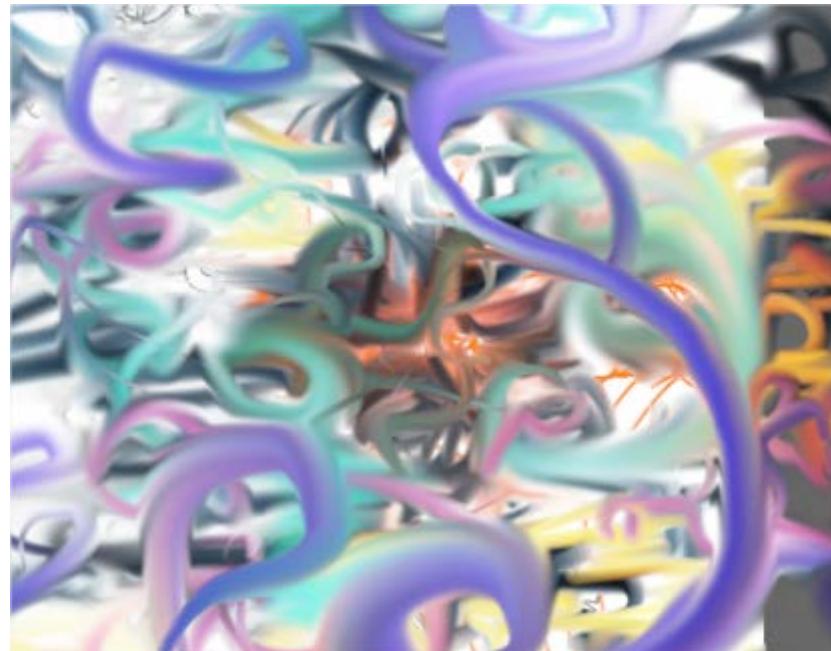
```



```

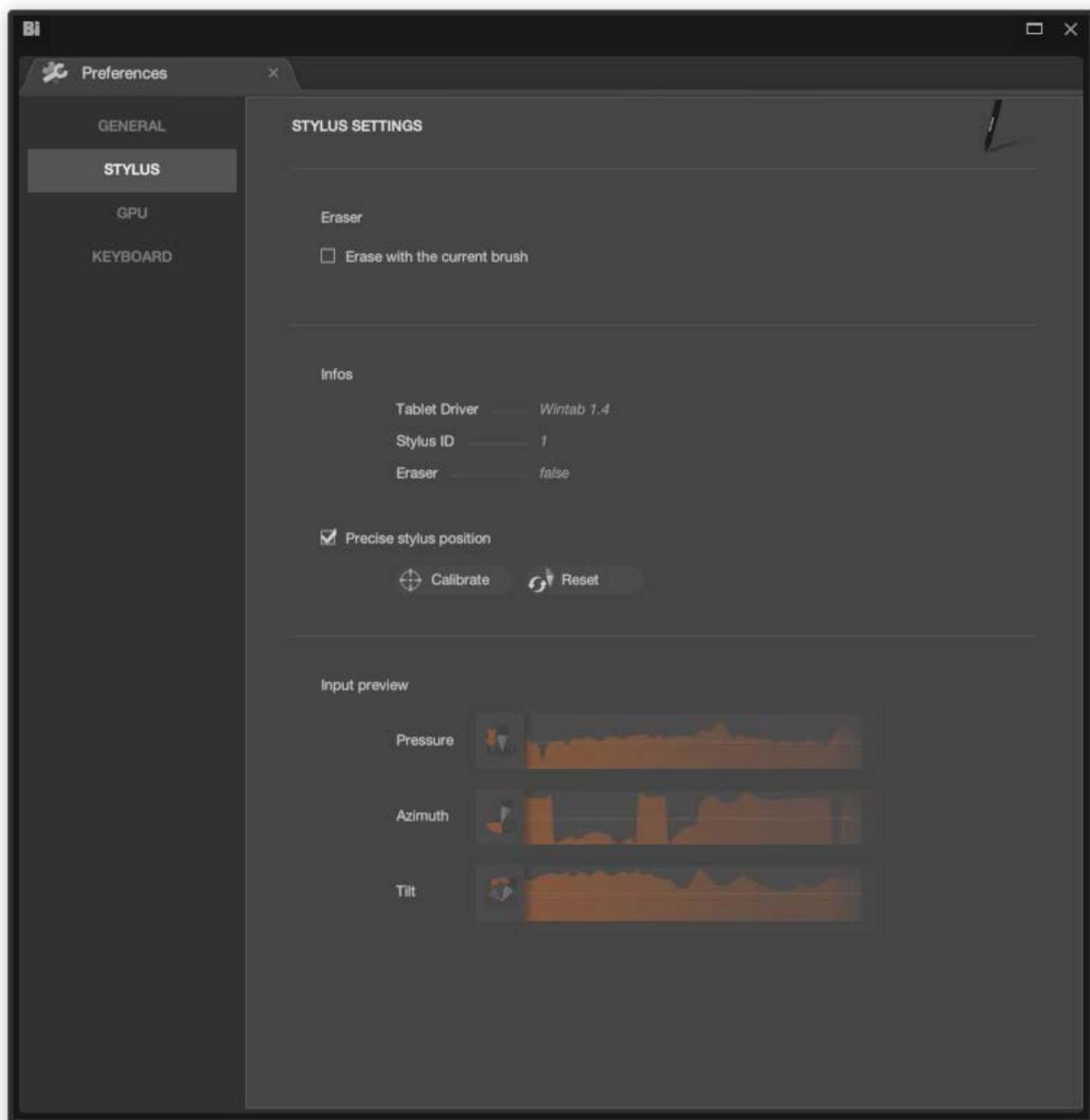
float4 main( idatas i )
{
    // get the current background color
    float4 colBack = bottomLayer.pointSample( i,
i.pos );
    // compute smudge factor
    float sfact = smudgeFactor( i );
    // compute displacement from the current
position
    float mdisplace = sfact * min(
samplingLayerMaxOffset, fromDrawSpace(
i.maxDisplace ) );
    float2 poff = -i.dir * mdisplace;
    // Get the displaced color
    float4 smpl = bottomLayer.bilinearSample( i,
i.pos + poff );
    smpl = smoothLerp( colBack, smpl, saturate(
mdisplace ) ); // reset to a pointSampling for
little displacement
    // apply a color overlay
    float4 tmp = blendOverlay( smpl,
i.color*float4(1,1,1,i.overlayFactor*pow(sfact,1.9
5)) );
    smpl.xyz = tmp.xyz;
    // blend
    float4 col = blendNormalAlpha( colBack, smpl,
i.color.a*pow(sfact,0.75) );
    return col;
}

```



Tutorial: Impostazione della tavoletta Wacom con BlackInk

Per vedere come reagisce la vostra tavoletta durante l'utilizzo di BlackInk occorre selezionare dal menu principale **Windows -> Preferences** e poi cliccare su **STYLUS**:



Oltre alle informazioni sul tipo di driver utilizzato, troviamo anche la visualizzazione grafica di come variano i parametri della tavoletta quando la utilizziamo: **Pressione (Pressure)**, **Azimuth** e **Tilt (Inclinazione)**.

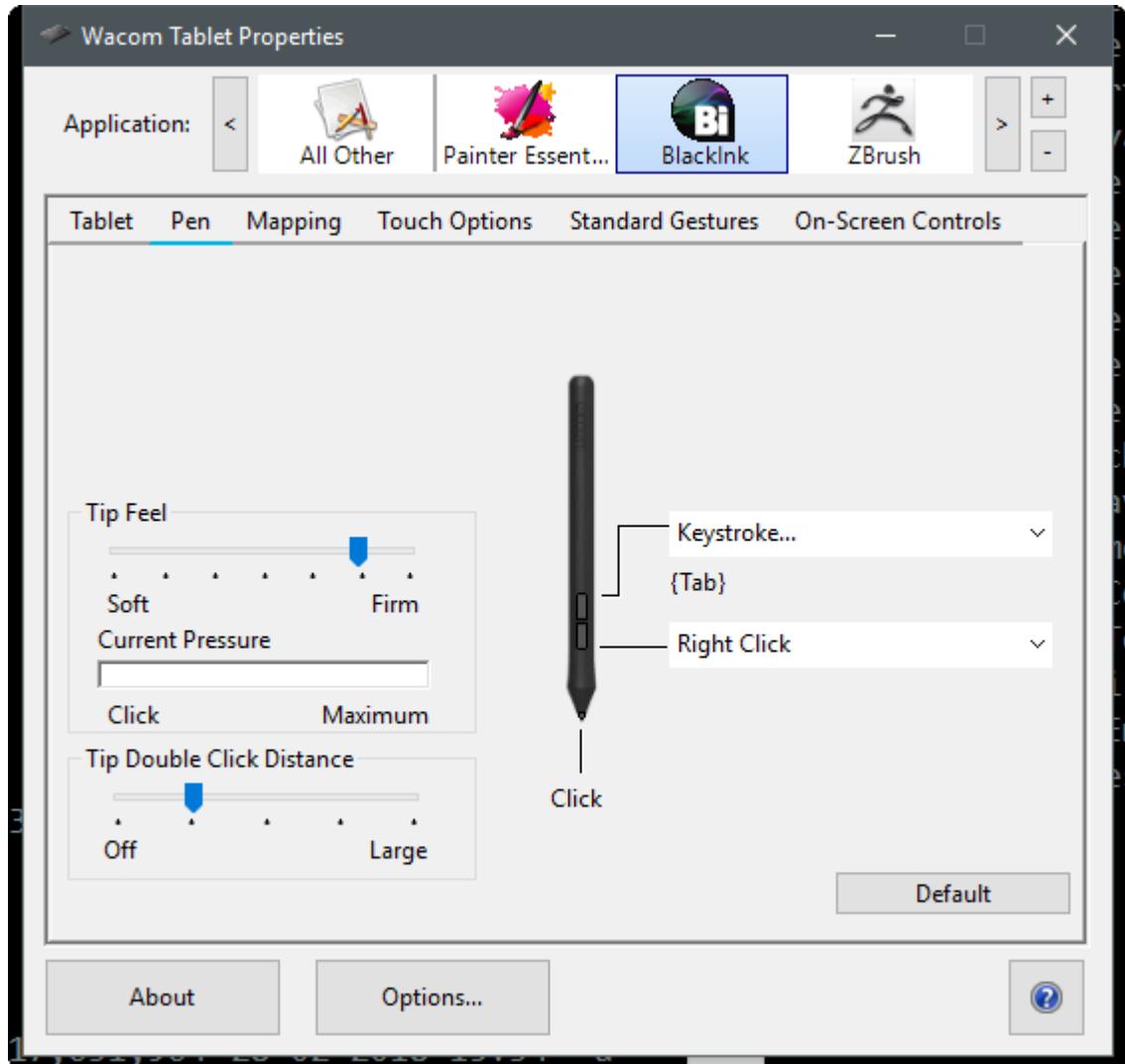
Attenzione: le tavolette più economiche non hanno il Tilt e/o l'Azimuth.

Se vogliamo impostare l'Opacità del tratto in funzione della Pressione della tavoletta dobbiamo effettuare due operazioni:

- 1) Regolazione della Pressione utilizzando il driver della tavoletta

2) Regolazione dell'Opacità utilizzando i controller di BlackInk

1) Regolazione della Pressione utilizzando il driver della tavoletta Wacom

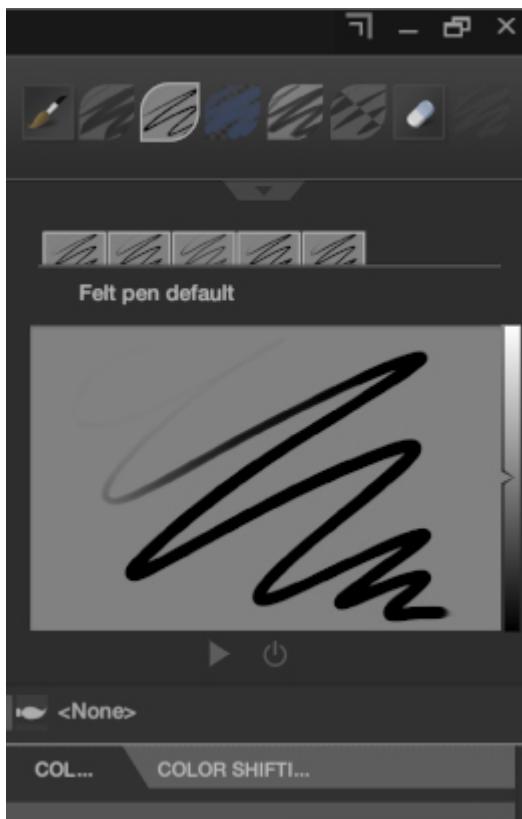


Da questa finestra potete regolare tutti i parametri della vostra penna.

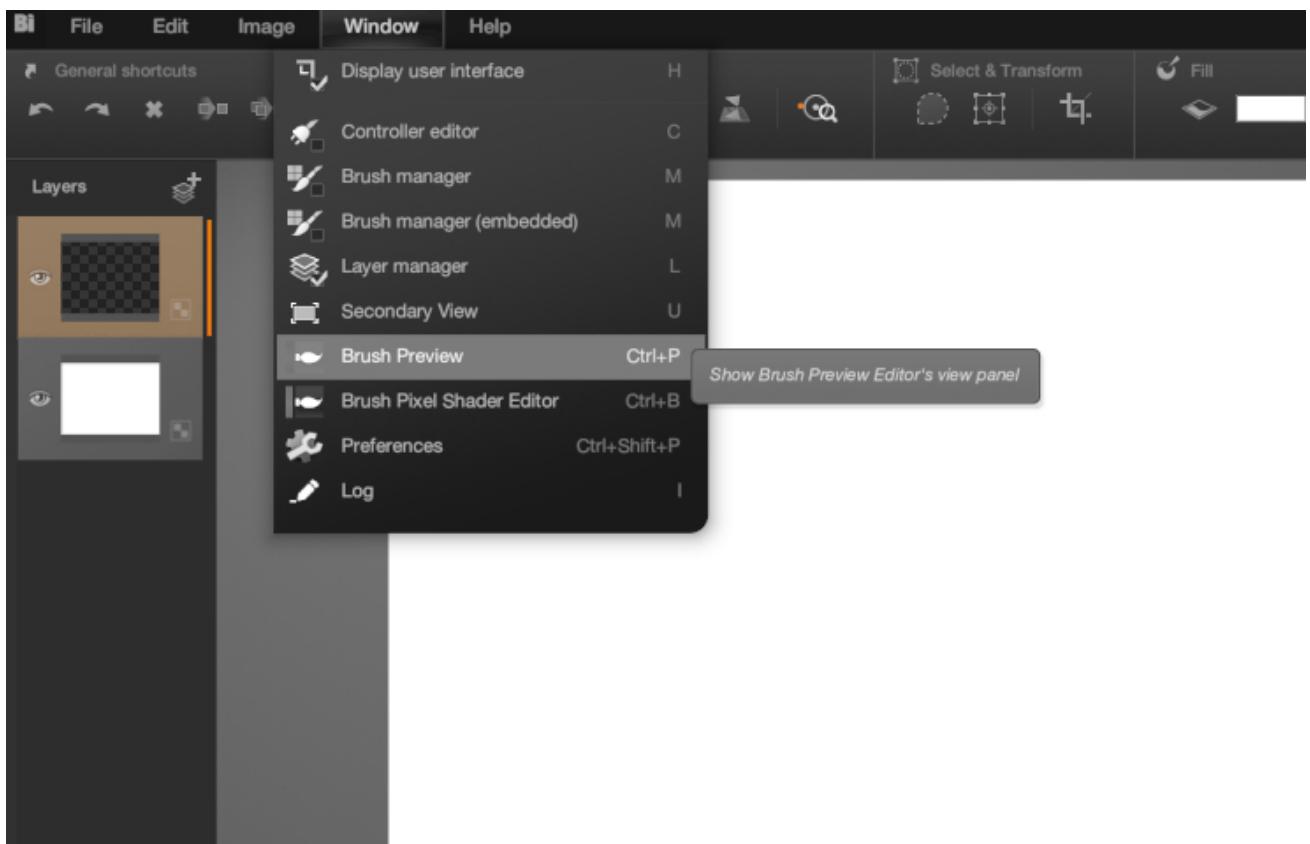
2) Regolazione dell'Opacità utilizzando i controller di BlackInk

Adesso passiamo a regolare la pressione con i controller.

Avviate BlackInk e scegliete il pennello **Felt pen default**:



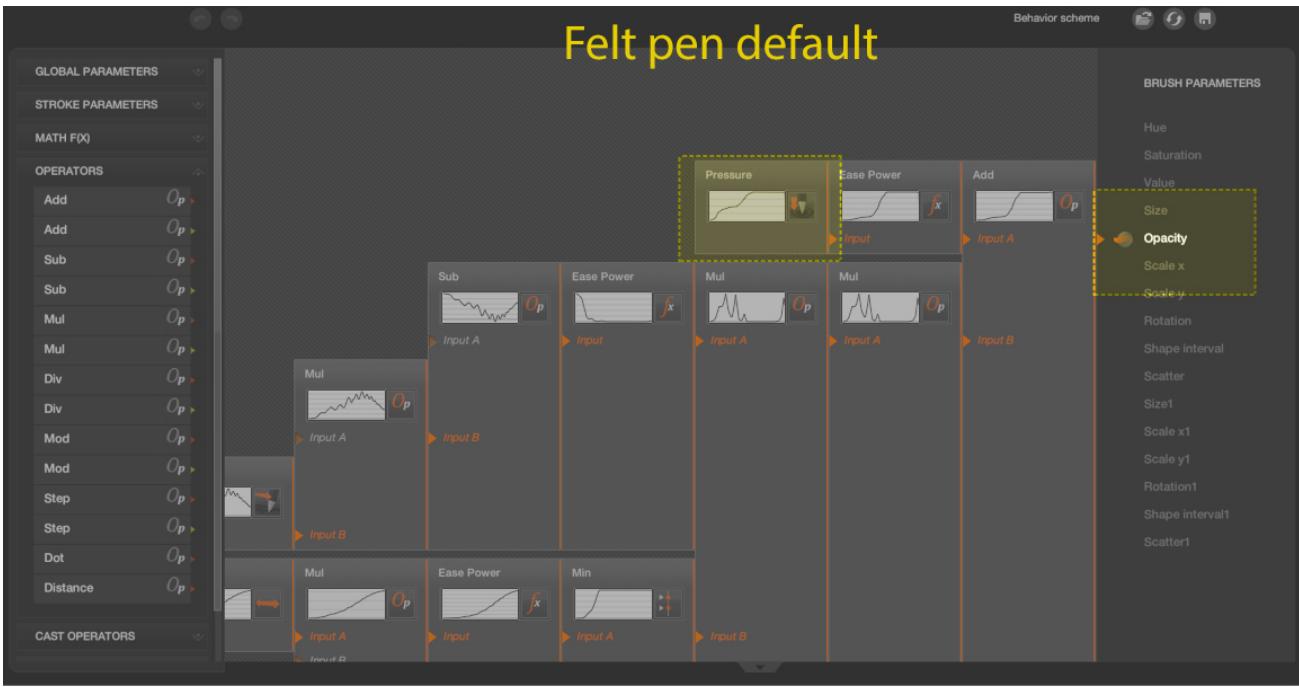
Poi apriamo la finestra **Brush Preview** (Anteprima Pennello) (Ctrl+P):



Disegniamo qualcosa come riferimento:



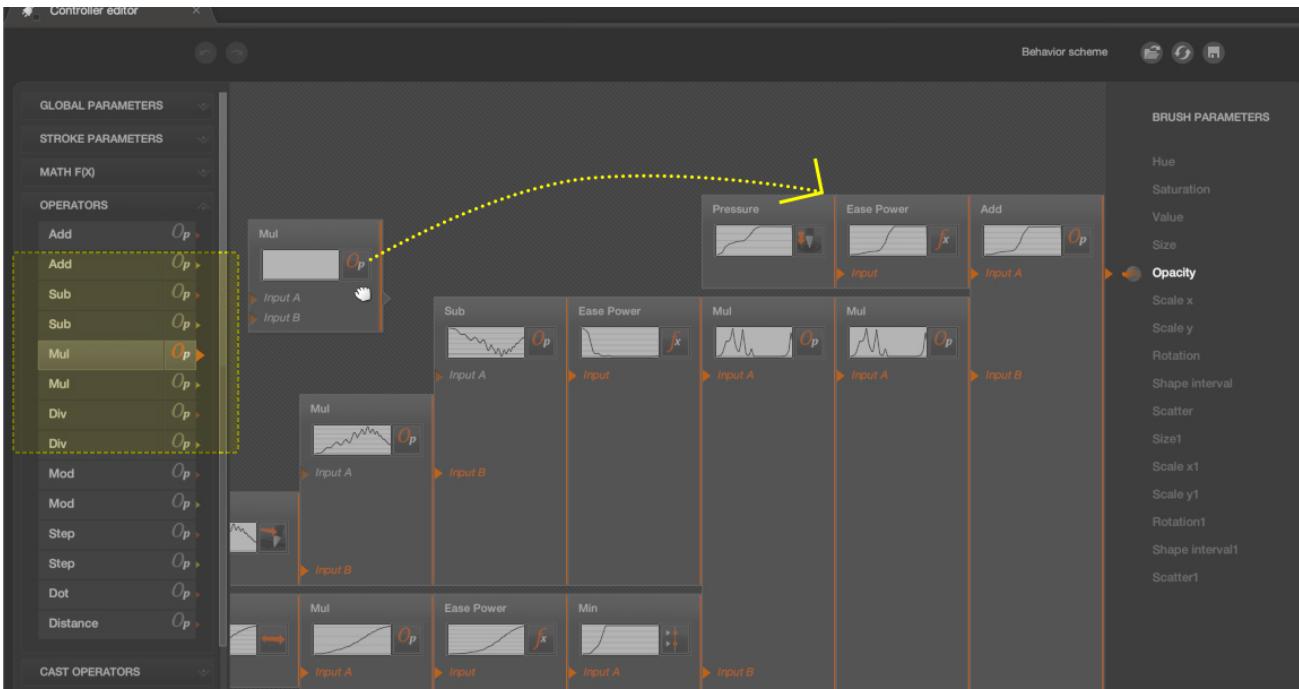
Adesso apriamo il pannello del **Controller Editor** (C) e scegliamo il parametro **Opacity (Opacità)**:



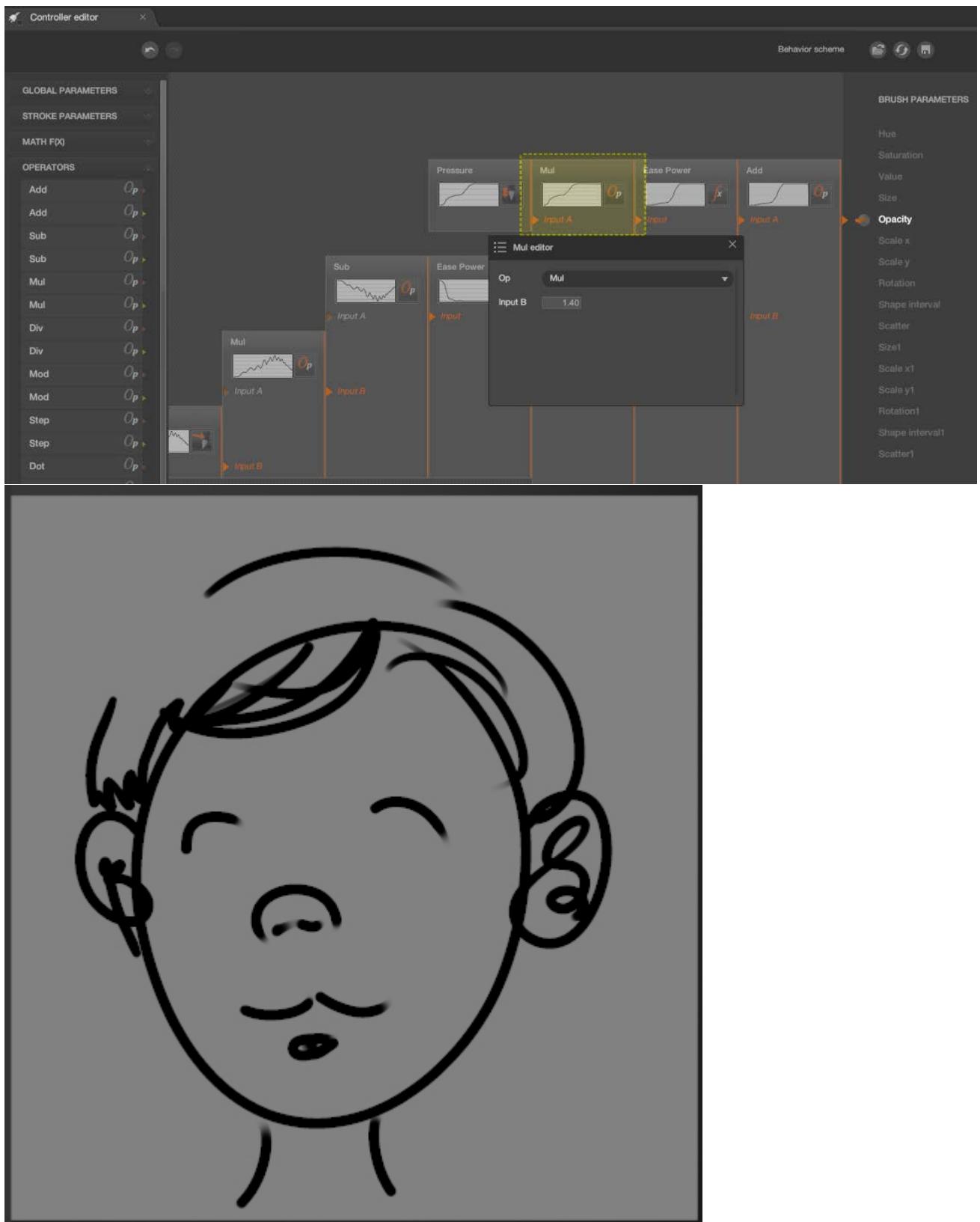
Applichiamo un moltiplicatore per modificare la sua intensità.

Prendi il nodo **Mul** dallo slot **OPERATORS** e inseriscilo tra il nodo **Pressure** (**Pressione**) e il nodo **Ease Power**.

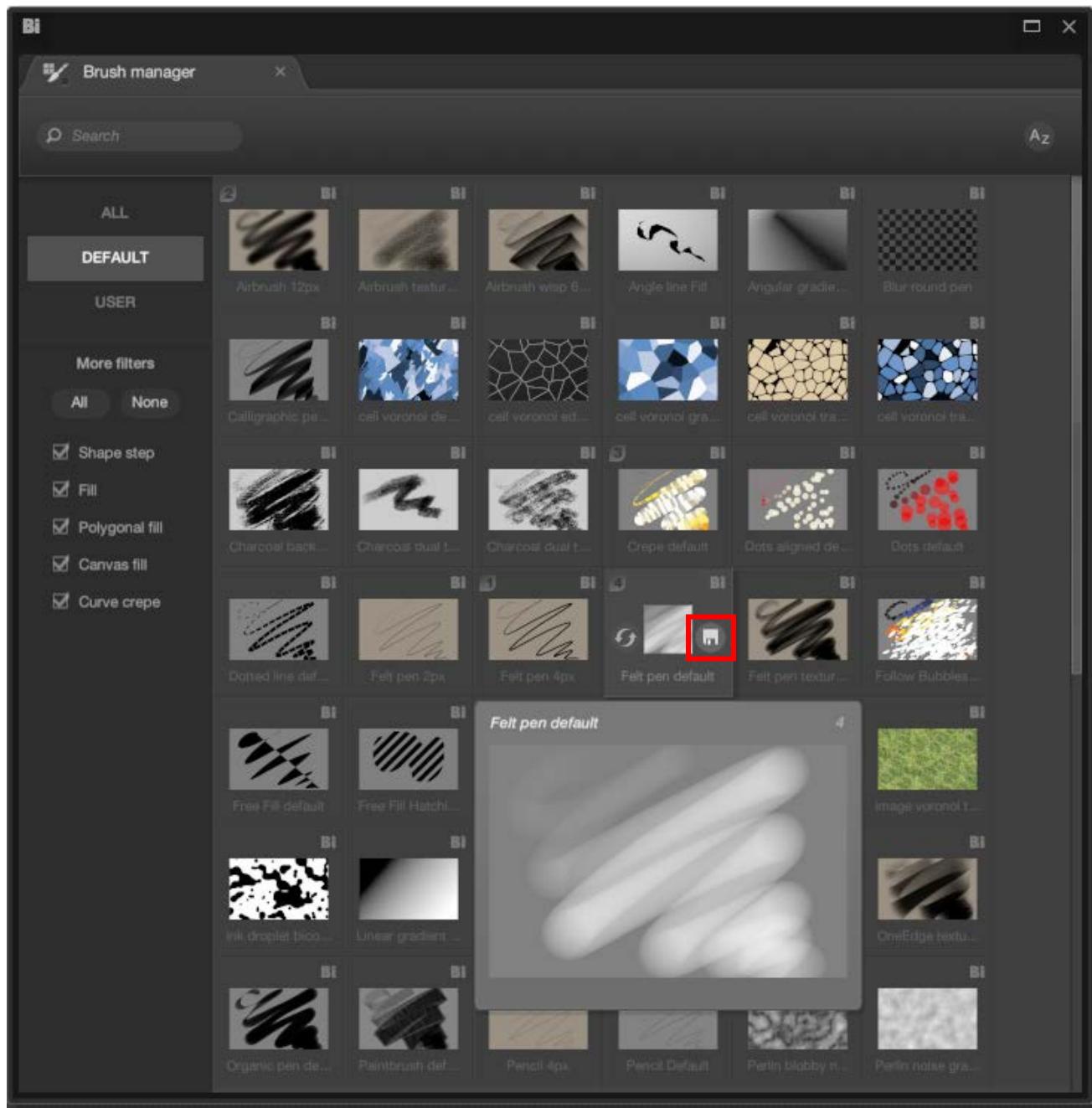
Per fare questo devi prima scollegare il nodo **Pressure**, collegare il nodo **Mul** al nodo **Ease Power** e poi collegare il nodo **Pressure** sull'ingresso Input A del nodo **Mul**:



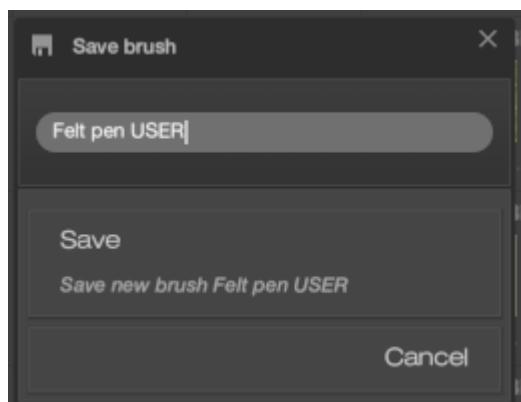
Per regolare in modo preciso l'opacità del tratto variamo il valore di **Input B** del nodo **Mul**. Durante questa regolazione osservate come cambiano i tratti del disegno nella finestra **Brush Preview** (**Anteprima Pennello**) (fate alcune prove per trovare il valore che vi soddisfa):



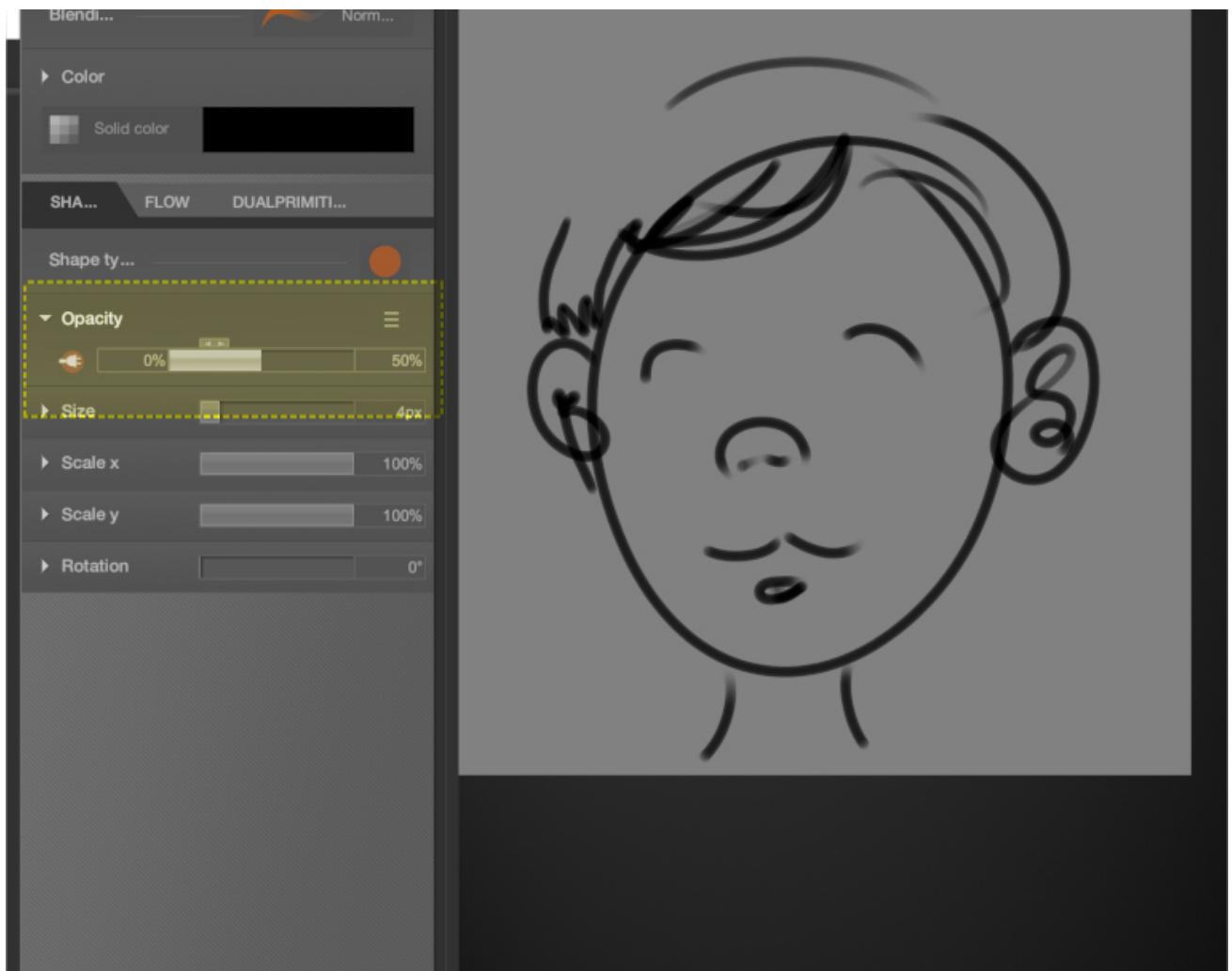
Una volta terminato puoi salvare il nuovo pennello aprendo il **Brush Manager (M)** e poi premendo il pulsante indicato dal riquadro rosso:



Inserite un nuovo nome e premete il tasto Save:



Naturalmente puoi impostare altri parametri in funzione della Pressione, ma puoi anche modificare il range dei valori come in questo esempio:



Questo è tutto.

NOTA:

Se la vostra tavoletta non è sensibile all'inclinazione (Tilt), allora non avrete la possibilità di gestire il nodo corrispondente (Tilt).

Controller Reference

Global Parameters

Canvas Size (Vec2)

Canvas size in pixels

ViewSize (Vec2)

Current viewport size in pixels

CanvasToView Zoom (Float)

Size difference between canvas and view.

Returns the current zoom value.

Global StartTime (Float)

Time in second since BlackInk's launch

Global StrokeID (Float)

The time in seconds since the BlackInk launch of a specific brush

Stroke Parameters

Pressure (Float)

Returns the tablet's pen pressure.

Nears 0 when no pressure is applied and goes to 1 at full pressure.

Speed (Float)

Returns the stroke's speed.

The speed is in pixel / second and then is normalized to [0, 1] from the range [0, 800 px/s]

Attaining a speed of 1 requires a very fast movement so you may want to multiply it by for example 5 to get an easier effect.

Time (Float)

Time elapsed since the start of the stroke.

It goes above 1 very quickly so you might need to divide it by 10 or 20 for direct time driven use. Is a good input for sinus or noise.

Length (Float)

Length of the stroke. Just like the time it attains 1 very quickly. Also a good input for sinus or noise.

Tilt (Float)

The altitude of the tablet's pen inclination.

Returns 1 when the pen is perpendicular from the tablet's surface and nears 0 when it is tilted very close to the surface.

Azimuth (Float)

The direction in which the tablet's pen is tilted, in degrees.

0 is when the pen is oriented towards the top of the tablet. Most common use is to divide it by 360 to obtain a 0 to 1 value.

Azimuth (Float)

Returns the azimuth angle (stylus orientation on the tablet) in radians.

Azimuth [0..1] (Float)

Returns the azimuth angle (stylus orientation on the tablet) as a unit vector.

Direction (Float)

The direction of the stroke in radians (not degrees).

0 is going towards the left of the document. Like the azimuth, you will most of the time want to divide it by 2π . For example, to make a shape follow the direction of the stroke, divide direction by 2π , plug it in the rotation parameter and set the brush rotation from 0 to 360.

Direction (Vec2)

Stroke direction as a vector in canvas space

Direction [0..1] (Float)

Normalized stroke direction in canvas space

View Direction (Float)

Stroke direction in view space

View Direction (Vec2)

Stroke direction in view space

View Direction [0..1] (Float)

Normalized stroke direction in view space

Direction Perpendicular (Vec2)

Unit vector perpendicular to the stroke direction in canvas space

Pos (Float)

Current position (X and Y) on the document, in pixels, from the top left corner.

Pos [0..1] (Float)

Current position on the document, normalized, from the top left corner.

X returns 0 on the left of the document and 1 on the right, Y 0 at the top and 1 at the bottom.

View Pos (Float)

Current position (X and Y) in BlackInk's drawing viewport, in pixels, from the top left corner.

View Pos [0..1] (Float)

Current position (X and Y) in BlackInk's drawing viewport, normalized, from the top left corner.

Primitive ID (Float)

The number of the current primitive since the beginning of the stroke.

Math F(X)

Ease Cubic (Float)

Simple ease function to smooth the ends of a value.

Ease Power (Float)

Power function.

The value is multiplied by itself a number of times defined in the block's parameters. The result is a value staying closer to zero unless it nears 1. Useful with pressure to gain more control over low pressures and keep the highest values for when really pressing intentionally on the tablet.

Easy Elastic (Float)

Sinus (Float)

Returns a wave effect from an input: the value regularly goes up and down between -1 and 1.

The block's parameters are predefined to return a result between 0 and 1, for easier use. Time or Length are good inputs because they both have constantly increasing values.

Abs (Float)

Returns the absolute value of the input. -0.5 returns 0.5, 0.5 returns 0.5.

Abs (Vec2)

Returns the absolute value of the input. -0.5 will return 0.5, 0.5 will return 0.5.

Floor (Float)

Returns the nearest lower integer. 0.5 returns 0, 1.5 returns 1, 2.5 returns 2.

Floor (Vec2)

Returns the nearest lower integer. 0.5 returns 0, 1.5 returns 1, 2.5 returns 2.

Ceil (Float)

Returns the nearest higher integer. -1.1 will return -1, 0 will return 0, 1.2 will return 2, 2.8 will return 3, and so on.

Ceil (Vec2)

Returns the nearest higher integer. -1.1 will return -1, 0 will return 0, 1.2 will return 2, 2.8 will return 3, and so on.

Frac (Float)

Returns the fractional value of the input. 0.5 returns 0.5, 1.5 returns 0.5, 2.5 returns 0.5.

Frac (Vec2)

Returns the fractional value of the input. 0.5 returns 0.5, 1.5 returns 0.5, 2.5 returns 0.5.

Exp (Float)

Exponential function.

Log (Float)

Logarithmic function.

Perpendicular (Vec2)

Returns a vector perpendicular to the input vector.

Swap (Vec2)

Swaps the X and Y values of a vector.

Normalize (Vec2)

Normalizes a vector.

Operators

Add (Float)

Adds input A and input B.

Add (Vec2)

Adds input A and input B.

Sub (Float)

Subtracts input B to input A.

Useful to invert values: plug the input value in B and set A to 1. For example a pressure in opacity will make the shape more opaque when you press harder and 1-Pressure will do the opposite, making it more opaque when you press less.

Sub (Vec2)

Subtracts input B to input A.

This can be used easily invert an value. Plug the input value into B and set A to 1. For example a pressure in opacity will make the shape more opaque when you press harder and 1 - Pressure will do the opposite, making it more opaque when you press less.

Mul (Float)

Multiplies input A and input B.

Very easy operation to use: for example, multiplying a value by the pressure will (at low pressure) multiply by a number closer to 0, making it smaller. At full pressure, it will multiply by 1, not changing it.

Mul (Vec2)

Multiplies input A and input B.

Div (Float)

Divides input A by input B.

Div (Vec2)

Divides input A by input B.

Mod (Float)

Modulo. Remainder of the division of input A by input B.

In practical terms, it allows to repeat a value from zero to a defined limit. For example length with a modulo at 0.5 will go up as usual but when it attains 0.5, will restart from 0.

Mod (Vec2)

Modulo operator. The remainder of the division of Input A by input B. Allows a value to increment from 0 to a defined limit, and then restart from 0 when it reaches the limit

Step

Returns 0 if input A is inferior to B, 1 if superior.

In other terms, the B value is used to choose whether the A input is converted to a 0 or a 1. For example: If B is set to 0.5, 0.1 returns 0, 0.4 returns 0, 0.6 returns 1, 0.9 returns 1.

Step (Vec2)

Returns 0 if Input A is lower to B, 1 if greater. It is a bit like a boolean check. If B is set to 0.5, A at 0.1 will return 0 as it's lower. If A is 0.6, it will return 1 as it's greater.

Dot (Float)

Dot product operation between two vectors. Returns 1 when vectors are pointing in the same direction, 0 when they are perpendicular and -1 when pointing in the opposite direction.

Distance (Float)

Returns the distance between two vectors.

Cast Operators

AsVec2 (Vec2)

Creates a vector2 with the input value as X and Y.

ToVec2 (Vec2)

Combines inputs A and B into a vector2.

ToVec2Normalize (Vec2)

Combines inputs A and B into a normalized vector2.

GetX (Float)

Returns the X value of a vector.

GetY (Float)

Returns the Y value of a vector.

GetAngle (Float)

Returns a vector's angle in radians.

GetAngle [0..1] (Float)

Returns the normalized angle (between 0 and 1) of a vector in radians.

GetLength (Float)

Returns the length of a vector.

FromSpherical (Vec2)

Creates a vector from an angle (float) and a length (float).

Signals

Smooth (Float)

Smooth the input value by factor.

Smooth (Vec2)

Smooths the input value by a factor.

Random (Float)

Generate a random value between 0 and 1. It has no input. Fun for creating a brush with unpredictable features.

Random Disk (Vec2)

Generates a uniform random value within a circle with a radius of 1 as a vector2.

Random Square (Vec2)

Generates a uniform random value from -1 to 1 on both X and Y as a vector2.

Noise Smooth (Float)

Generate a smooth noise from the input value. Time or length are often used for most common regular results.

Noise Smooth (Vec2)

Generates a smooth noise from the input value. Time or length are often used with this for regular results.

Noise Cell (Float)

Generate a square noise from the input value. The value jumps from one result to another without interpolation.

Noise Cell (Vec2)

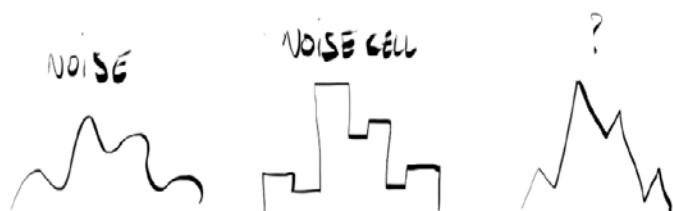
Generates a square noise from the input value. This causes the value to jump from one result to other with no interpolation (Smoothing).

Noise Triangle (Float)

Generates an angular noise.

Noise Triangle (Vec2)

Generates an angular noise.



MIXING

Mix (Float)

Blends input A and input B by the factor input.

When the factor value goes towards 0 the result tends to be the value given in A and when going towards 1, the one in B. A factor of 0.5 is the average of the two inputs.

Mix (Vec2)

Blends input A and input B by the factor input.

Min (Float)

Returns the lowest value between input A and input B.

Min (Vec2)

Returns the lowest value between input A and input B.

Max (Float)

Returns the highest value between input A and input B.

Max (Vec2)

Returns the highest value between input A and input B.

Clamp (Float)

Limits input value to min and max.

Every value below min become the min and every value above max becomes the max. Note that once plugged in the final output, the parameters only consider values between 0 and 1 so there's no need to clamp before use.

Clamp (Vec2)

Limits input value between a min and a max.

Every value below min will become the min, and every value above will become the max. Once plugged into the final output the parameters only consider values between 0 and 1.

Tasti scorciatoia (Shortcuts)

COMMON	
Ctrl+N	New
Ctrl+Shift+N	New... (Creates popup and gives you control over canvas size and resolution)
Ctrl+O	Open
Ctrl+S	Save (Save the current document)
Ctrl+Shift+S	Save As
Ctrl+F4	Quit
Ctrl+Z	Undo
Ctrl+Y	Redo
Ctrl+C	Copy
Ctrl+Shift+C	Copy Merged
Ctrl+X	Cut
Del	Delete the active layer content
Ctrl+V	Paste
Esc	Exit
NAVIGATE	
<i>To get back to a default view, hover the navigation icons in the main toolbar to reveal their options.</i>	
Hold Space	Pan/Move around the canvas
Hold R	Rotate the canvas
Shift+R	Snap rotation
Ctrl+R	Remove Rotation
Hold Q	Zoom in and out
Ctrl+1	Zoom to 100%
Ctrl+0	Fit canvas to screen
+	Zoom in (Step Zoom in)
-	Zoom out (Step Zoom Out)
VIEW	
F	Flip Horizontal (Drawing view is flipped horizontally)
Shift+F	Flip Vertical (Drawing view is flipped vertically)
Y	Constant screen brush size (Brush size depends on view zoom level, it's constant with it)
PAINT	
General	
Hold S	Change the size of your brush moving the mouse or stylus
Hold Z	Set the drawing smoothness
Hold D	Change the color opacity of your brush moving the mouse or stylus
Brush	
Num 1-5	Switch between your favourite brushes
B	Come back to your previous brush
Hold Shift	Constrain your brush as straight line when you are drawing
K	Toggle Straight line mode
Y	Toggle viewconstant brush size
Eraser	
Hit E	Switch between the eraser/brush

Hold E	Erase with the current brush.
<i>That means that you can now erase with any brush you like!</i>	
Color Picker	
Hold Alt	Quickly pick a color (This works only when using the solid color mode)
FILL	
Hit G	Fill layer / selection
Ctrl+G	Fill layer alpha
Ctrl+Alt+G	Fill transparency
Fill eyedropper	
Alt+G	Quickly pick a fill color
SELECTION	
General	
W	Start selection
Ctrl+Shift+A	Select Current Layer Opacity
Ctrl+A	Select All
Ctrl+I	Invert selection
Ctrl+D	Deselect
Shift+Draw	Preserve Ratio 1:1 (Rectangle and ellipse selection only)
Operations	
Shift+Draw	Set Add operation
Alt+Draw	Set Sub operation
Shift+Alt+Draw	Set Intersection Operation
Adjustment	
Shift+Click	Add a point to the line (polygonal lasso only)
Shift+Click	Delete a point (polygonal lasso only)
Ctrl+T	Transform latest selection
TRANSFORMATION	
T	Transform gizmo
Hold Shift	Constrain axis displacement and snap angle
Shift+C	Crop Tool
SHOW	
H	Toggle user interface
Ctrl+←	Left panel layout
Ctrl+↓	Bottom panel layout
Ctrl+→	Right panel layout
Panel	
C	Toggle brush control editor
L	Toggle layer manager
M	Toggle Brush Manager panel
U	Show secondary view panel
I	Show log view panel

Shortcuts BlackInk

Command Name	Shortcut
New	Ctrl+N
New..	Ctrl+Shift+N
Open..	Ctrl+O
Save	Ctrl+S
Save as..	Ctrl+Shift+S
Exit	Ctrl+F4
Undo	Ctrl+Z
Redo	Ctrl+Y
Toggle controller editor	C
Toggle layer manager	L
Toggle secondary view panel	U
Toggle log view panel	I
Toggle brush manager panel	M
Preferences	Ctrl+Shift+P
Brush Preview	Ctrl+P
Brush Pixel Shader Editor	Ctrl+B

Toggle Black Ink User Interface	H
Toggle left panel layout	Ctrl+Left Arrow
Toggle bottom panel layout	Ctrl+Down Arrow
Toggle right panel layout	Ctrl+Right Arrow
UI Scaling	 Ctrl+Q
Cut	Ctrl+X
Copy	Ctrl+C
Copy merged	Ctrl+Shift+C
Paste	Ctrl+V
Clear	Del
Resize..	Ctrl+R
Crop Tool	Shift+C
 Enable snapping	 Shift
Fit on screen	Ctrl+Num 0
100% zoom	Ctrl+Num 1
Zoom In	Num +
Zoom Out	Num -

Canvas view move		Space
Center view		Ctrl+Space
Zoom in and out		Q
Rotation Tool		R
Remove rotation		Ctrl+Shift+R
90° counter clockwise		Shift+Alt+R
90° clockwise		
Toggle flip horizontal		F
Toggle flip Vertical		Shift+F
Toggle Constant screen brush size		Y
Brush Favorite 1		Num 1
Brush Favorite 2		Num 2
Brush Favorite 3		Num 3
Brush Favorite 4		Num 4
Brush Favorite 5		Num 5
Select Previous Brush		B
Toggle the eraser		E

Toggle the eraser	E
Erase with the current brush.	 E
Fill all	G
Fill Alpha	Ctrl+G
Fill invert Alpha	Ctrl+Alt+G
Fill eye dropper	Alt+G
Smoothness	 Z
Set Brush Size	 S
Set Brush Opacity	 D
Straight line draw mode	K
Transform Tool	T
Transform selection	Ctrl+T
Selection tool	W
Select All	Ctrl+A
Select alpha	Ctrl+Shift+A
Invert selection	Ctrl+I
Deselect	Ctrl+D
Color eye dropper	 Alt

Pennelli predefiniti (ordine alfabetico)

