



# Full Circle

LE MAGAZINE INDÉPENDANT DE LA COMMUNAUTÉ UBUNTU LINUX

ÉDITION SPÉCIALE SÉRIE INKSCAPE



ÉDITION SPÉCIALE  
SÉRIE INKSCAPE



## INKSCAPE

Volume Huit Parties 50 à 57

Full Circle Magazine n'est affilié en aucune manière à Canonical Ltd.



Spécial Full Circle Magazine

# Full Circle

LE MAGAZINE INDÉPENDANT DE LA COMMUNAUTÉ UBUNTU LINUX

## BIENVENUE DANS UN AUTRE « NUMÉRO SPÉCIAL »

Une autre série, une autre compilation d'articles pour plus de commodité. Voici une réédition directe de la série Inkscape, articles 50 à 57 des numéros 110 à 117. Du propre aveu du non-artiste Mark Crutch : s'il peut le faire, vous le pouvez aussi !

Veillez garder à l'esprit la date de publication originale ; les versions actuelles du matériel et des logiciels peuvent être différentes de celles illustrées. Il convient donc de vérifier la version de votre matériel et de vos logiciels avant d'essayer d'émuler les tutoriels dans ces numéros spéciaux. Il se peut que les logiciels que vous avez installés soient plus récents ou qu'il y ait des versions plus récentes disponibles dans les dépôts de votre distribution.

**Amusez-vous !**

## Au sujet du Full Circle

Le Full Circle est un magazine gratuit, libre et indépendant, consacré à toutes les versions d'Ubuntu, qui fait partie des systèmes d'exploitation Linux. Chaque mois, nous publions des tutoriels, que nous espérons utiles, et des articles proposés par des lecteurs. Le Podcast, un complément du Full Circle, parle du magazine même, mais aussi de tout ce qui peut vous intéresser dans ce domaine.

## Clause de non-responsabilité :

Cette édition spéciale vous est fournie sans aucune garantie ; les auteurs et le magazine Full Circle déclinent toute responsabilité pour des pertes ou dommages éventuels si des lecteurs choisissent d'en appliquer le contenu à leurs ordinateur et matériel ou à ceux des autres.

## Sommaire

<b>Partie 50 :</b>	page 3
<b>Partie 51 :</b>	page 7
<b>Partie 52 :</b>	page 11
<b>Partie 53 :</b>	page 15
<b>Partie 54 :</b>	page 19
<b>Partie 55 :</b>	page 22
<b>Partie 56 :</b>	page 26
<b>Partie 57 :</b>	page 30
<b>Écrire pour le FCM :</b>	page 34
<b>Mécènes :</b>	page 35
<b>Comment contribuer :</b>	page 36



Les articles contenus dans ce magazine sont publiés sous la licence Creative Commons Attribution-Share Alike 3.0 Unported license. Cela signifie que vous pouvez adapter, copier, distribuer et transmettre les articles mais uniquement sous les conditions suivantes : vous devez citer le nom de l'auteur d'une certaine manière (au moins un nom, une adresse e-mail ou une URL) et le nom du magazine (« Full Circle Magazine ») ainsi que l'URL [www.fullcirclemagazine.org](http://www.fullcirclemagazine.org) (sans pour autant suggérer qu'ils approuvent votre utilisation de l'œuvre). Si vous modifiez, transformez ou adaptez cette création, vous devez distribuer la création qui en résulte sous la même licence ou une similaire.

**Full Circle Magazine est entièrement indépendant de Canonical, le sponsor des projets Ubuntu. Vous ne devez en aucun cas présumer que les avis et les opinions exprimés ici ont reçu l'approbation de Canonical.**



D'abord, mille excuses. Le mois dernier, j'ai suggéré que vous pourriez colorer une ombre noire en utilisant la colonne Décalage fixe du filtre Matrice de couleurs, et je l'ai démontré en utilisant l'entrée Opacité de la source. Malheureusement, une modification a été introduite dans la version 0.91 d'Inkscape qui empêche les décalages fixes des composantes de couleur à partir de l'opacité de la source (<https://bugs.launchpad.net/inkscape/+bug/897236>). Ça marche dans la version 0.48, comme dans Firefox et les autres moteurs de rendu SVG. Mes excuses à tous ceux qui ont perdu leur temps en essayant de suivre mes instructions avec la 0.91, et merci à **Moini** sur le forum Inkscape d'avoir porté ce problème à mon attention. Bon, poursuivons...

Un autre type d'effet d'ombre portée que vous verrez de temps en temps est « l'ombre empilée ». Elle est créée en empilant l'une sur l'autre plusieurs copies à bords nets de votre objet original, chacune d'une couleur de remplissage différente.

La manière facile de créer cet effet est de simplement dupliquer vos objets

originaux, de les déplacer un petit peu, de changer leur couleur de remplissage et de les rempiler dans le bon ordre. Avec trois objets, celui du milieu n'ayant ni remplissage, ni trait, il suffit de quelques instants pour produire ceci :

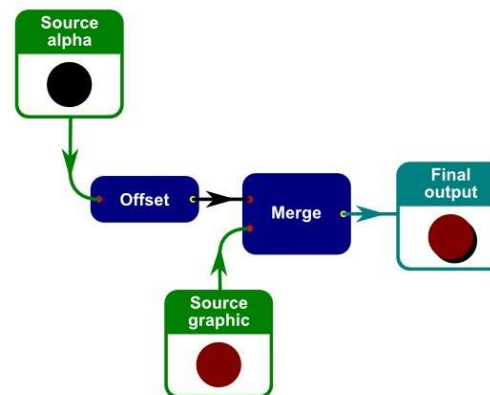
## STACKED SHADOW

Ce n'est pas un mauvais début, mais que se passe-t-il si vous avez besoin de modifier le texte ? Vous devrez le modifier dans les trois objets, ce qui triple vos chances de faire une erreur. Il vaudrait mieux utiliser des remplissages indéfinis avec des clones (voir partie 30), qui vous permettraient d'obtenir le même résultat, mais avec un seul objet parent à modifier pour propager vos modifications sur toute la pile.

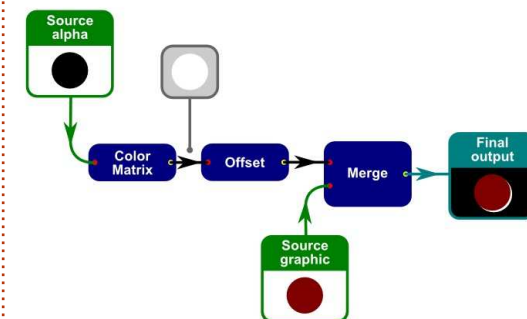
Cependant, même avec les clones, vous travaillez toujours avec trois objets. Un regroupement vous permet de les bouger ou de les transformer d'un seul coup, mais, alors, vous avez

le travail supplémentaire d'entrer dans le groupe et d'isoler l'objet original pour modifier le texte. Comme vous pourrez l'imaginer, les filtres offrent une solution à tous ces problèmes.

Avec les quelques primitives de filtrage que nous avons apprises dans les deux articles précédents, vous en savez déjà assez pour créer un effet d'ombre empilée en utilisant une couleur de remplissage au sommet, avec une copie blanche de l'image source en-dessous d'elle et une copie noire tout en bas. C'est vraiment identique à une simple ombre portée à bords nets (voir partie 48) avec une ombre portée recolorée (partie 49) en sandwich. Regardons-la d'abord sous forme de graphe, en commençant par une ombre portée de base noire avec des bords nets :



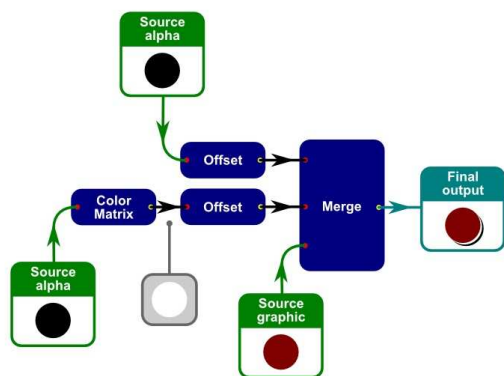
Tout simple, non ? Maintenant, regardons notre ombre portée blanche à bords nets. Vous remarquerez que c'est pratiquement le même graphe, mais avec l'ajout d'une primitive Matrice de couleurs pour convertir en blanc l'ombre noire (j'utiliserai un fond noir pour la boîte de sortie finale, pour que l'ombre blanche soit visible) :



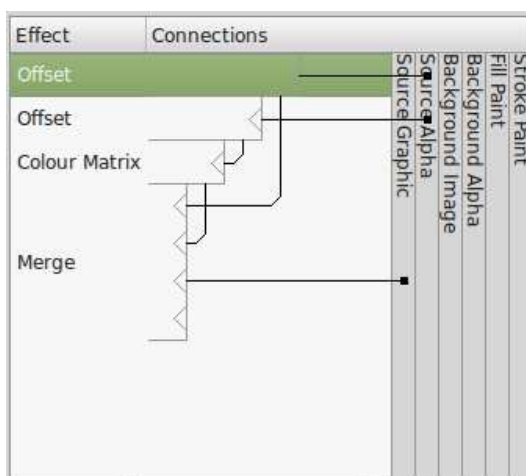
Chaque ligne R, G et B doit valoir au moins 1,0 (ce qui équivaut à 255 en RGB), de façon à obtenir une sortie blanche avec la matrice de couleurs. Nos valeurs d'entrée sont toutes à zéro, de sorte qu'aucune multiplication ne donnera le résultat que nous voulons. À la place, nous devons mettre une valeur de 1,0, ou plus, dans la colonne de décalage fixe des trois premières lignes :

1.00	0.00	0.00	0.00	1.00
0.00	1.00	0.00	0.00	1.00
0.00	0.00	1.00	0.00	1.00
0.00	0.00	0.00	1.00	0.00

Maintenant que vous savez comment créer les éléments constitutifs de notre filtre, nous devons juste les combiner. Dans le cas présent, ce n'est qu'une question de fusion dans le bon ordre : l'ombre noire d'abord, puis la blanche et enfin la source image. Le graphe final ressemble à ceci :



Comme vous pouvez le voir, notre filtre final a besoin de quatre primitives : deux Décalages, une Matrice de couleurs et une Fusion. Il a aussi deux connexions à l'entrée Opacité de la source et une à l'entrée Source image. Regardons la conception finale du filtre dans Inkscape :



Si vous suivez chaque ligne de l'image, vous verrez que c'est le même ensemble d'objets connectés que dans le graphe. Malheureusement, l'interface d'Inkscape rend son aspect plus complexe, principalement du fait que les lignes se croisent pour que toutes les branches du filtre arrivent sur la primitive Fusion. Maintenant, imaginez le même dessin de filtre, mais en empilant plus d'ombres : bien que chaque ombre soit un graphe linéaire

séparé se connectant sur une Fusion commune, l'interface d'Inkscape devient vite remplie d'un méli-mélo de lignes croisées. Si vous vous trouvez confronté à un tel bazar, essayez de dessiner les primitives de filtrage et leurs connexions sous forme de graphe pour voir si ça devient plus compréhensible.

Il y a un petit problème avec notre filtre d'ombres empilées : le résultat paraît différent suivant qu'il est placé sur un fond blanc ou coloré. Dans le premier cas, l'ombre blanche se dissout dans le fond, donnant l'impression d'une ombre noire disjointe, mais dès que nous la mettons sur un fond différent, la couche blanche se détache.

Dans certains cas, nous pouvons souhaiter que la couche blanche soit visible, mais, dans d'autres, nous préférions préférer que cette partie de la

sortie soit transparente. Si vous aviez construit les ombres empilées à partir d'objets SVG standard, vous pourriez utiliser un chemin de découpe pour réaliser cet effet (voir partie 13), mais les chemins de découpe ne sont pas disponibles comme primitives de filtrage. À la place, il y a une primitive appelée « Composite » qui vous permet de combiner deux entrées de différentes façons, y compris deux ou trois ayant le même effet qu'un chemin de découpe.

La primitive Composite utilise les valeurs d'opacité des pixels des images d'entrée pour déterminer ce que sera le pixel de sortie, utilisant les méthodes décrites par Thomas Porter et Tom Duff dans les années 80, connues sous le nom de modes fondus Porter-Duff. Ces modes fondus sont sélectionnés depuis la liste déroulante Opérateur de l'éditeur de filtres :

**Défaut** - Ceci supprime l'opérateur de la primitive de filtrage dans le fichier XML sous-jacent. D'après la spécification des effets de filtrage SVG, Inkscape est ainsi amené à se comporter comme si une valeur de « Over » avait été fournie. Pour plus de clarté, je vous recommande de ne jamais utiliser cette option, et de toujours choisir l'option « Over » si c'est ce que vous voulez.

**Over** - Les deux images sont l'une sur l'autre, avec l'entrée du haut apparaissant





sant au-dessus de celle du bas. C'est exactement la même chose qu'en utilisant la primitive Fusion avec deux entrées, sauf que l'ordre des deux entrées est inversé.

**In** - Seules les parties de l'image du haut qui sont à l'intérieur de l'image du bas apparaîtront à la sortie. L'effet est le même qu'avec un chemin de découpe.

**Out** - Seules les parties de l'image du haut qui sont à l'extérieur de l'image du bas apparaîtront à la sortie. L'effet est le même qu'avec un chemin de découpe « inversé ».

**Atop** - La sortie se compose de l'image d'entrée du dessous, plus toutes les parties de l'image du haut qui sont à l'intérieur de l'image du bas.

**XOR** - Réalise une opération OU exclusif sur les valeurs RGB de chacun des pixels des images d'entrée. L'effet est l'inclusion dans l'image de sortie de toutes les parties des images d'entrée qui ne se recouvrent pas.

**Arithmetic** - Ce n'est pas un des modes fondus Porter-Duff, mais plutôt l'ajout d'un mode présent dans la spécification SVG. Je le décrirai prochainement un peu plus en détail.

Notez que l'interface utilisateur du filtre fournit 4 réglettes, mais, même si elles ne servent que pour l'opérateur Arithmetic, elles restent néanmoins visibles, bien que désactivées, lorsqu'un

autre de ces opérateurs est utilisé.

Les descriptions ci-dessus sont en gros correctes, mais certaines subtilités apparaissent quand les images d'entrée contiennent des valeurs d'opacité autres que 0 et 255. Si vous voulez utiliser cette primitive pour une découpe, il est donc préférable de vérifier que les images d'entrée ne contiennent pas de valeurs intermédiaires. La meilleure façon de le faire est avec la primitive Transfert de composantes, qui a acquis une interface utilisateur dans la version 0.91 et sera décrite dans un futur article. En se limitant aux filtres que j'ai déjà décrits, vous pouvez aussi utiliser la primitive Matrice de couleurs pour élargir et décaler la plage des valeurs possibles afin d'obtenir le même résultat. Par exemple, cette matrice resserrera les valeurs d'opacité, de sorte que celles qui sont inférieures à 128 seront converties en 0, et celles égales ou supérieures seront remplacées par 255.

1.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	512.00	-256.00

Comme tant de choses dans SVG, il y a un tas de mots pour décrire ce qui est plus compréhensible avec une

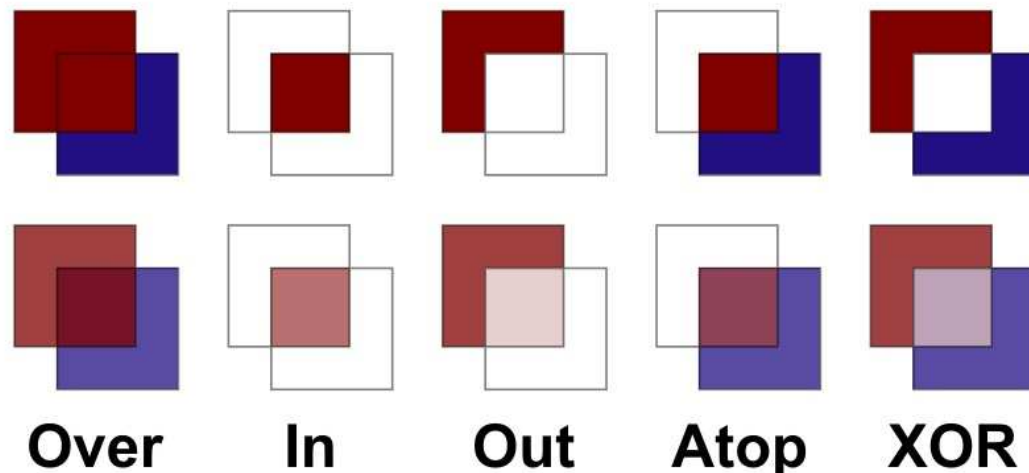


image. Voici les 5 modes fondus Porter-Duff quand ils sont appliqués à une paire de carrés, d'abord sans transparence, puis avec une opacité réduite à 75%. Notez que les lignes noires d'entourage ont été ajoutées après coup pour souligner les parties restantes des images - elles ne sont pas présentes dans la vraie sortie de filtrage.

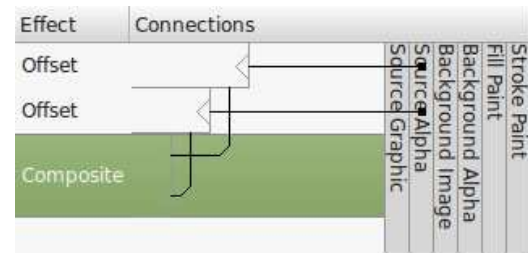
Revenons à notre ombre empilée et regardons comment ce filtre peut nous aider à éliminer la couche blanche. Considérez simplement une petite partie de la sortie : une lettre unique. J'ai retiré l'image source si bien que nous ne voyons que les deux ombres décalées (col. 4).

Nous devons garder la partie noire qui est visible, mais enlever tout le



contenu blanc, le laissant transparent. En d'autres termes, nous voulons garder la partie de la couche noire qui est extérieure à la couche blanche. C'est évidemment un travail pour le mode fondu « Out » de la primitive Composite. Comme le filtre Composite ne s'intéresse qu'à l'opacité des sources d'entrée, pas à leur couleur, nous pouvons nous passer de la primitive Matrice de couleurs, ce qui nous donne

comme résultat cette chaîne de filtrage :



Tout ce dont nous avons besoin est d'ajouter à la fin de la chaîne un bloc pour fusionner encore une fois cette sortie avec l'image source originale, nous donnant le résultat final que nous cherchions - une ombre empilée avec une couche intermédiaire transparente qui fonctionne sur n'importe quel fond (voir en haut, au milieu).

Il y a une dernière chose à décrire avant de conclure l'article de ce mois : le mode « Arithmetic » de ce filtre



Composite et ses quatre réglettes (K1 à K4). Avec ce mode, les quatre canaux (R, G, B, A) de chaque pixel de l'image de sortie sont calculés à partir du canal du pixel correspondant des images d'entrée (i1 et i2), pondérés par les valeurs K1 à K4 en utilisant la formule suivante :

$$\text{résultat} = (K1 \times i1 \times i2) + (K2 \times i1) + (K3 \times i2) + K4$$

En décomposant ceci, vous pouvez voir que K4 est multiplié par rien ; il représente donc un décalage fixe. K2 et K3 sont respectivement multipliés par i1 et i2 ; ainsi, ils ajustent la quantité de chaque entrée qui compose la sortie. K1 est multiplié par i1 et i2 à la fois, si bien qu'il élargit la plage des valeurs de sortie.

Ce mode peut être utilisé pour combiner la sortie de deux autres primitives de filtrage, vous permettant de contrôler les proportions de chaque entrée. La spécification SVG suggère qu'il pourrait être utile pour superposer la sortie de certaines des primitives des effets d'éclairage (non encore décrites dans cette série) avec les données de texture d'une autre primitive ou d'une image source, mais il peut être utile si vous voulez mélanger deux images tout en contrôlant la force apportée par chacune.

## FORUM INKSCAPE

Le conseil d'administration d'Inkscape forme un comité en vue d'organiser la création d'un forum officiel sur Inkscape. Le principal forum existant dans la communauté ([inkscapeforum.com](http://inkscapeforum.com)) est devenu une cible pour les spammeurs et le propriétaire du domaine ne répond plus aux mails et aux messages. Le président de ce comité sera Brynn, un contributeur de longue date sur l'ancien site, qui entretient un autre forum sur [www.inkscapecommunity.com](http://www.inkscapecommunity.com). Les principaux contributeurs du forum passent sur son site, au moins à titre provisoire. Jusqu'à ce qu'une décision finale soit prise à propos du nouveau forum, il est fortement recommandé que les messages de soutien et les demandes soient postés sur le site de Brynn, plutôt que sur l'ancien forum.



**Mark** a utilisé Inkscape pour créer trois bandes dessinées, *The Greys*, *Monsters*, *Inked* et *Elvie*, qui peuvent toutes être trouvées à <http://www.peppertop.com/>



Il apparaît que l'annonce du mois dernier à propos des forums fut un peu précipitée. Il suffit de dire que les comportements et les personnalités étaient en opposition dans le monde des forums de soutien Inkscape, mais que, depuis, les choses se sont calmées !

Aussi, je vais m'en tenir aux faits : les deux forums inkscapeforum.com et inkscapecommunity.com continuent à fonctionner, chacun avec un sous-ensemble des utilisateurs (et un peu de recouvrement). D'une manière générale, les demandes de support posées sur l'un d'eux auront droit à une réponse et les utilisateurs ordinaires d'Inkscape n'ont pas à s'inquiéter des manigances de coulisse. Rien de tout ça n'a d'impact sur le développement d'Inkscape lui-même. Quand des informations éventuelles, plus concrètes, se feront jour à propos d'un forum officiel, je vous en parlerai, mais, jusque-là, je garderai les jeux politiques des forums en dehors de ces colonnes !

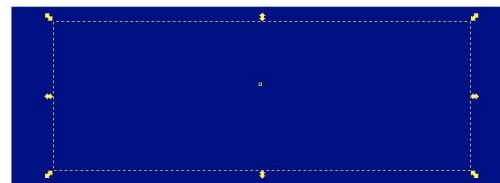
**B**on, où en étions-nous ? Ah, oui, aux filtres. Revenons à l'ombre portée monochrome de la partie 49 pour vous montrer une façon plus simple de créer cet effet. Précédemment,

j'avais parlé de la primitive Matrice des couleurs comme un moyen de convertir une couleur en une autre, mais, quand tout ce qu'il y a à faire est d'introduire une seule couleur fixe dans votre chaîne de filtres, il est généralement plus simple d'utiliser la primitive Remplissage.

Comme vous pouvez le deviner par son nom, la primitive Remplissage inonde une zone avec une couleur. Vous êtes peut-être en train de penser à des trucs comme l'outil Pot de peinture dans Inkscape ou d'autres programmes graphiques, qui généralement remplissent une zone en travaillant du centre vers le bord jusqu'à ce qu'il rencontre une bordure d'une autre couleur. Mais il n'y a pas autant de finesse ici ; la primitive Remplissage remplit la totalité de la « région des effets du filtre » avec une couleur unie. La région des effets du filtre est le rectangle défini dans l'onglet Paramètres généraux des filtres (voir partie 48) et est typiquement plus grande que la boîte englobante des objets sélectionnés.

En recommençant avec un texte simple, créez un filtre et ajoutez la primitive

Remplissage. La primitive étant sélectionnée, utilisez les contrôles en bas du dialogue des effets de filtres pour choisir une couleur et une opacité, et vous devriez obtenir un résultat comme celui-ci (notez que l'entrée à laquelle le filtre est connecté n'a pas d'importance, car elle n'a pas d'effet sur la sortie) :



Pas vraiment exaltant, n'est-ce pas ? Aussi, la question, maintenant, est de savoir comment transformer ce grand rectangle bleu en ombre portée aux formes douces. Si vous avez suivi le tutoriel du mois dernier, vous savez que le filtre Composite (en mode « In ») peut être utilisé pour découper le rectangle bleu à la forme de votre texte.



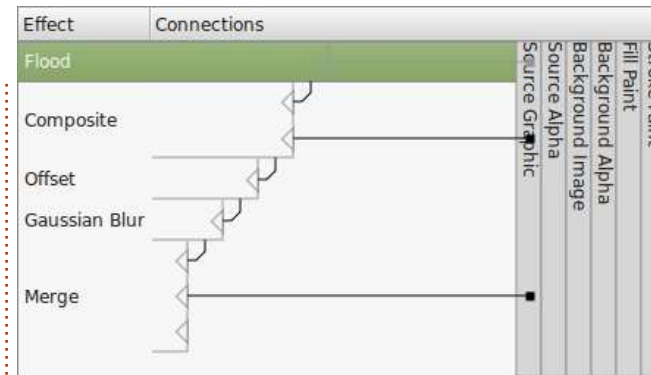
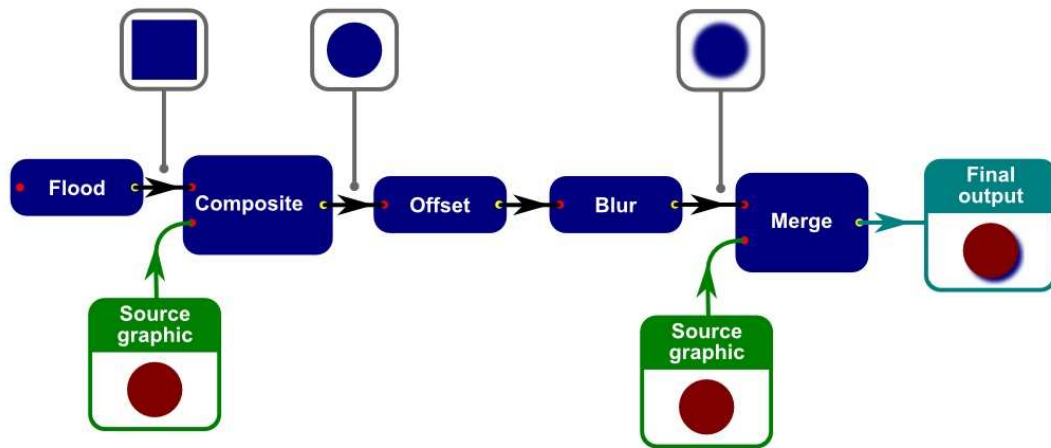
Ce n'est pas très classique d'avoir une sortie de filtre bien rectangulaire ; vous voudriez plutôt qu'il suive les formes et les courbes de vos objets sélectionnés. Ainsi, si vous voyez une primitive Remplissage dans votre chaîne de filtres, il y a de bonnes chances qu'un filtre Composite le suive de près pour l'ajuster à la forme.

Maintenant que nous avons obtenu une version colorée du texte, il y a une manière directe de le décaler et de le flouter, avant de le fusionner avec la Source image ; vous devriez être habitué à ces étapes maintenant et je vous épargnerai une description détaillée. À la place, je présenterai les résultats, une vision graphique de la chaîne et une copie d'écran d'Inkscape.



Alors que Remplissage vous fournit un rectangle d'une seule couleur, la primitive Turbulence vous donne un rectangle rempli d'un mélange chaotique de couleurs. Ce n'est pas vrai-



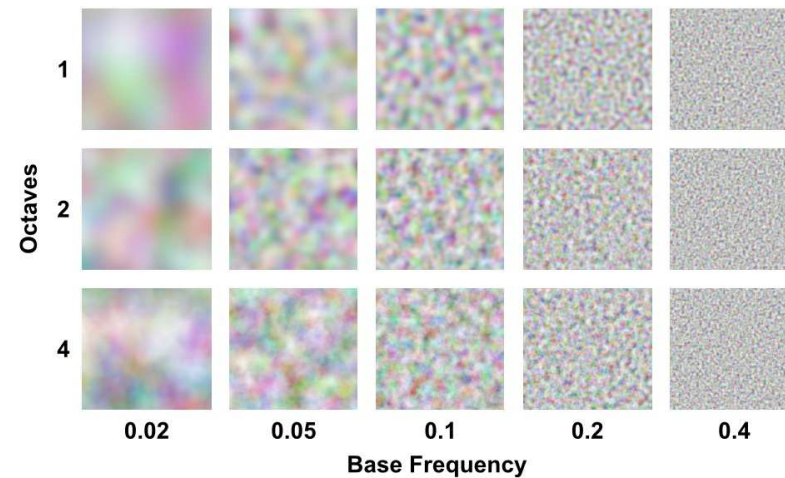


ment aléatoire, au sens mathématique du terme, car la sortie est bien définie et reproductible (cela signifie que tous vos filtres se ressembleraient quel que soit l'outil de rendu, ce qui n'est pas toujours le cas en réalité), mais, dans un sens courant, c'est vers cette primitive que vous devez aller si vous voulez ajouter un degré d'aléa ou de bruit à votre image. Elle a deux modes : Bruit fractal et Turbulence. La différence entre les deux est que le dernier a plus de « creux » dans la sortie, où le fond transparait, donnant l'apparence de lignes réunies partout dans la sortie, alors que le premier a plus l'apparence d'un nuage.

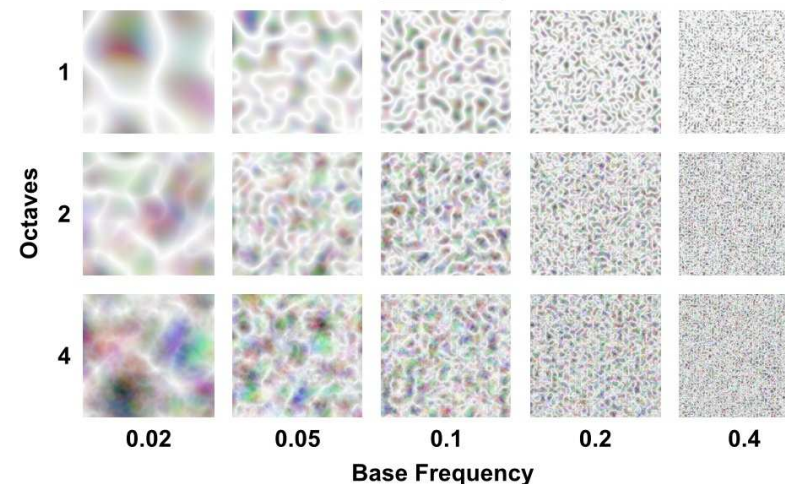
Quel que soit le mode choisi, le reste des contrôles est le même. Les réglettes de la Fréquence de base contrôlent la « densité » affichée du bruit, les basses valeurs en donnent

peu, avec des transitions douces, alors qu'avec de valeurs plus élevées, les changements sont plus rapides dans les transitions, donnant à la sortie un aspect « neigeux » qui rappellera les vieilles télévisions non réglées. Les fréquences horizontale et verticale ont le plus souvent la même valeur, mais peuvent être modifiées indépendamment en appuyant sur le bouton Lien sur la droite. La réglette Octaves contrôle la précision ou la complexité du bruit ; la pousser au-delà de 4 ne vaut que rarement la peine, car l'augmentation du détail est trop petite pour être visible et cela impose une charge supplémentaire au processeur. Enfin, la valeur du Germe peut être utilisée pour alimenter le générateur de nombre pseudo-aléatoire au cœur du filtre pour avoir un motif de sortie légèrement différent sans changer les autres paramètres.

**Fractal Noise**



**Turbulence**





Les images suivantes montrent l'effet de différentes valeurs des réglages Fréquence de base et Octaves, pour les modes Bruit fractal et Turbulence.

Vous aurez remarqué que les images sont d'un ton nettement pastel. C'est parce que les quatre canaux (R, G, B, A) sont calculés indépendamment - chaque pixel consiste réellement en une combinaison de 4 nombres pseudo-aléatoires. La valeur du canal Alpha écrasera toutes les autres ; ainsi, même s'il vous arrive d'avoir une couleur forte dans les composantes RGB, un Alpha faible va la réduire à une pâle ombre de lui-même.

5.00	0.00	0.00	0.00	0.00
0.00	5.00	0.00	0.00	0.00
0.00	0.00	5.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00

Vous pouvez utiliser une Matrice des couleurs pour en extraire un seul canal, ou pour étirer la sortie pour la rendre plus animée. Dans cet exemple, j'ai appliqué cette dernière, ainsi que supprimé tout le canal Alpha en le remplaçant par une valeur fixe de 1 (complètement opaque). La couleur cyan du texte original ne transparait pas du tout dans ce cas (ce qui est cyan dans le résultat vient uniquement du filtre Turbulence), mais j'ai utilisé un fond noir pour faire plus ressortir les couleurs.

Pour extraire un seul canal de la sortie, mettez tout à zéro dans votre Matrice des couleurs et, ensuite, valorisez une seule des quatre premières colonnes, suivant ce que vous voulez obtenir en sortie. Par exemple, en mettant la valeur 1.00 dans chaque champ de la troisième colonne, les 0-255 valeurs de votre canal Bleu seront utilisées pour remplir les RGBA de la sortie. Vous pourriez vouloir sortir le canal Alpha de l'équation en mettant



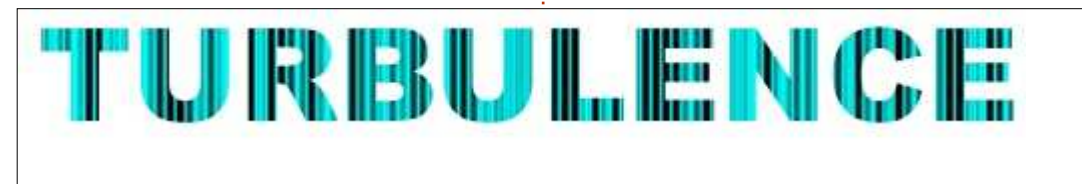
la valeur 1.00 dans le coin en bas à droite (la colonne Valeur fixe de la sortie Alpha). Dans cet exemple, j'ai utilisé le canal vert pour régler la sortie de l'Alpha, et étiré un peu les valeurs en utilisant 3.00 plutôt que 1.00. Cela donne une image qui va du noir opaque au noir transparent ; aussi, en la composant (pour l'attacher à la forme) puis en la fusionnant avec la Source image cyan, il est facile de créer un effet « électrique » ou « plasma ».

0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	3.00	0.00	0.00	0.00

Vous pouvez aussi ajouter une primitive Remplissage dans une chaîne comme celle-là pour vous assurer que votre résultat aura la bonne couleur dans le filtre, sans tenir compte de la couleur appliquée à l'objet.

N'oubliez pas que les contrôles de Fréquence de base peuvent être déliés. En gardant les valeurs proches l'une de l'autre, vous pouvez introduire un léger étirement ou décalage dans les motifs, alors qu'en les différenciant plus, des lignes assez horizontales ou verticales peuvent apparaître comme résultat. Voici le précédent filtre, mais avec la fréquence de base horizontale placée assez haut, et la fréquence de base verticale à zéro - le résultat n'est pas ce qu'on appelle habituellement « troublé », mais il peut néanmoins être une addition utile à votre arsenal de filtres.

Arrivés là, vous devrez commencer à percevoir la puissance et la flexibilité des filtres. En combinant quelques primitives de différentes manières, vous pouvez rapidement créer des résultats complexes. Mettez-y une pincée de chaos pseudo-aléatoire et tout vous est accessible, des nuages au marbre, alors que les primitives Rem-



# TUTORIEL - INKSCAPE

plissage peuvent garantir que les couleurs importantes dans votre filtre soient indépendantes des objets auxquelles elles sont appliquées.

Un thème commun entre Remplissage et Turbulence est qu'elles remplissent entièrement la région du filtre, nécessitant habituellement une opération Composite pour les remettre en forme.

Le mois prochain, nous examinerons la dernière primitive de « remplissage » - dévoilant en même temps les limitations d'Inkscape - puis progresserons dans les autres façons de modifier leur forme.



Mark a utilisé Inkscape pour créer trois bandes dessinées, *The Greys*, *Monsters*, *Inked* et *Elvie*, qui peuvent toutes être trouvées à <http://www.peppertop.com/>

## ÉDITIONS SPÉCIALES PYTHON :



<http://www.fullcirclemag.fr/?download/224>



<http://www.fullcirclemag.fr/?download/230>



<http://www.fullcirclemag.fr/?download/231>



<http://www.fullcirclemag.fr/?download/240>



<http://www.fullcirclemag.fr/?download/268>



<http://www.fullcirclemag.fr/?download/272>



<http://www.fullcirclemag.fr/?download/370>



<http://www.fullcirclemag.fr/?download/371>



<http://www.fullcirclemag.fr/?download/372>



<http://www.fullcirclemag.fr/?download/506>



<http://www.fullcirclemag.fr/?download/509>



<http://www.fullcirclemag.fr/?download/512>



La dernière fois, nous avons examiné Remplissage et Turbulence, deux primitives qui peuvent être utilisées pour remplir les zones de filtrage avec, respectivement, une couleur unie ou un nuage de couleurs pseudo-aléatoires. Mais il y a une foule d'autres remplissages que vous pourriez vouloir utiliser, des bandes jusqu'aux pois, des fleurs aux papillons. Pour répondre à ces possibilités infinies, la norme SVG fournit une méthode pour insérer une autre image dans votre chaîne de filtrage, en utilisant la primitive Image. Ceci vous permet, non seulement d'utiliser des images bitmap, mais peut aussi faire référence à d'autres parties de votre fichier SVG pour vous permettre d'insérer vos propres créations dans la chaîne de filtrage. Il n'y a qu'un petit problème : l'implémentation dans Inkscape est bel et bien inutilisable.

Commençons avec le peu qui fonctionne, au moins jusqu'à un certain point : l'importation d'une image bitmap externe dans votre chaîne de filtres. Comme d'habitude, nous commencerons avec un peu de texte comme objet auquel appliquer le filtre. Vous pouvez, bien sûr, utiliser n'importe quel type d'objet, mais je trouve que, plutôt qu'un simple rectangle ou cercle, un texte me permet de voir d'une façon rapide et facile ce à quoi ressemble le filtre quand il est appliqué à une forme complexe.

Créez un filtre sur l'objet test en utilisant une des méthodes décrites dans la partie 48 et, si nécessaire, enlevez toutes les primitives existantes. Maintenant, ajoutez une seule primitive Image à la chaîne de filtrage, et vérifiez que les contrôles sont au minimum en bas du dialogue. Le champ « Source de l'image » sera utilisé pour

déterminer le chemin et le nom de fichier du fichier image externe ou l'identifiant XML d'un autre élément de votre image. Pour l'instant, vous choisissez une image bitmap externe en cliquant sur le bouton « Fichier image » et la sélectionnant sur votre disque dur. Nous utiliserons notre image de *Mona Lisa* déjà essayée et adoptée, nous donnant la sortie suivante (texte non filtré à gauche, filtré à droite) quand le filtre est créé dans Inkscape 0.48 (en bas à gauche).

Vous n'avez vraiment pas besoin d'être un expert en art de la Renaissance pour voir que l'image est quelque peu déformée. Et voici le même filtre créé dans la 0.91 (en bas à droite).

Eh bien, nous avons perdu la déformation du format, mais la zone de filtre n'est pas vraiment remplie ; cependant, nous pouvons y remédier et

nous verrons cela sous peu. Ce changement de comportement pourrait éventuellement signifier que l'apparence des dessins créés dans la 0.48 pourrait ne pas être la même dans la 0.91 si vous utilisiez cette primitive de filtrage.

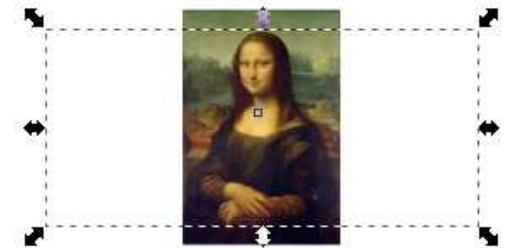
Dans la 0.48, la position par défaut et la taille de l'image nous sont imposées, c'est-à-dire étirées pour remplir la boîte englobante de l'objet. Le manuel officiel d'Inkscape laisse entendre que vous pouvez au moins régler la position et la taille de l'image dans le filtre en utilisant l'éditeur XML, mais, malgré tous mes essais, je n'ai pas réussi. Pour être honnête, le manuel dit que son implémentation dans Inkscape « ne positionne pas correctement les images », ce qui, d'après mes tests, est un euphémisme.

Avec la 0.91, c'est un peu mieux,

## IMAGE FILTER



## IMAGE FILTER





bien qu'il faille toujours passer par l'éditeur pour modifier les paramètres, car ils ne figurent toujours pas dans l'interface utilisateur. Replongez-vous dans la partie 31 de cette série si vous voulez rafraîchir votre mémoire au sujet de l'édition du XML. L'élément XML que vous devez modifier est `<svg:felimage>` qui est dans un `<svg:filter>` de la section `<svg:defs>` du fichier. Vous pouvez ajouter les attributs « x », « y », « width » (largeur) et « height » (hauteur), bien que, dans mes tests, aucun d'eux n'ait eu d'effet jusqu'à ce que j'ajoute aussi l'attribut « `preserveAspectRatio` » (préserver le rapport hauteur/largeur). Dans les tests suivants, une fois un tel attribut présent, j'ai dû changer sa valeur et retourner modifier les autres paramètres pour que ça soit pris en compte sur le canevas d'Inkscape. Si vous changez pour une valeur invalide et revenez à la valeur originale, cela fera l'affaire ; ainsi, il suffit de mettre une lettre à la fin de la valeur existante, cliquer sur « Définir », enlever la lettre et re-cliquer sur « Définir ».

Mais qu'est-ce qu'une bonne valeur pour `preserveAspectRatio` ? Un bon point de départ est le mot « none » (aucun), qui fera qu'Inkscape ignorera le rapport d'aspect de l'image originale et l'étirera pour remplir la boîte englobante de l'objet, le résultat ressemblant à celui de la version 0.48.

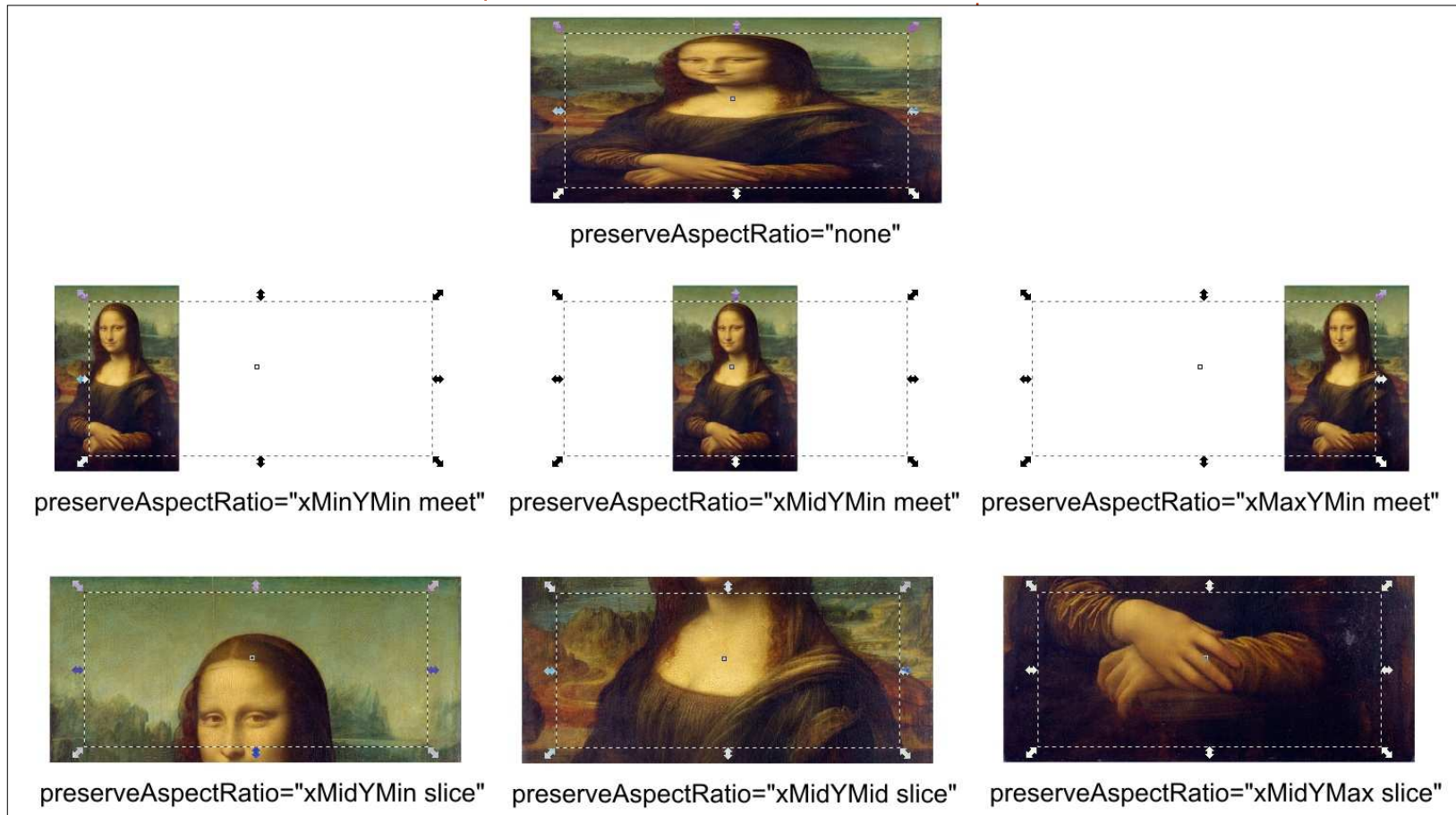
Mais il y a beaucoup d'autres options, toutes avec des noms similaires et déconcertants ! Elles commencent toutes par la lettre minuscule « x », immédiatement suivie de « Min », « Mid » ou « Max » (ce qui, pour la direction x, signifie aligné à gauche, centré ou aligné à droite), suivis par un « Y » majuscule et un autre « Min », « Mid » ou « Max » (aligné en haut, au milieu ou en bas), suivi d'un espace et du mot-clé optionnel « meet » (recadrer l'image pour qu'elle soit entièrement visible)

ou « slice » (mettre l'image à l'échelle pour remplir la boîte englobante, tout en préservant le ratio d'aspect, mais en cachant tout ce qui dépasse de la boîte englobante, c'est-à-dire, n'afficher qu'une tranche de l'image). Confus, n'est-ce pas ? Quelques exemples (ci-dessous) vont peut-être aider.

Non, je ne sais pas pourquoi le groupe de travail SVG en est arrivé à une similarité déroutante de « Min » et « Mid », pas plus que pourquoi « x »

est en minuscule alors que « Y » est en majuscule, ni pourquoi il a choisi les mots « meet » et « slice » plutôt que « scale » (échelle) et « crop » (rognier). Je sais bien, cependant, que mes exemples ne sont que la partie visible de l'iceberg : il y a 19 combinaisons possibles, sans considérer les « x », « Y », « width » et « height » déjà cités, qui peuvent avoir un effet important sur ce qui apparaît dans votre filtre.

Jusqu'à ce qu'Inkscape dispose d'une



interface utilisateur qui rende plus compréhensible toute cette folie, je recommande de laisser les options avancées de ce filtre à des experts. Mais si vous voulez déformer votre image pour qu'elle remplisse la boîte englobante, comme dans la 0.48, vous devrez prendre une grande respiration, remonter vos manches, et intervenir dans l'éditeur XML pour y mettre une rapide dose de :  
`preserveAspectRatio="none"`.

Il y a une dernière chose intéressante à noter à propos de la primitive Image, quand vous utilisez des images externes. Par défaut, Inkscape mettra le chemin complet vers votre image dans l'interface utilisateur. Pour rendre vos dessins plus portables, je vous recommande fortement de conserver les images nécessaires dans le même dossier que votre dessin, et, ensuite, de modifier manuellement la saisie des réglages du filtre pour enlever le chemin, ne laissant que le nom du fichier. Vous pourriez préférer incorporer votre image dans votre document, plutôt que la garder comme fichier externe, mais lisez la suite.

La primitive Image devrait avoir une astuce de plus dans son jeu, mais, une fois encore, c'est hors d'usage. Il est possible de sélectionner un objet (ou un groupe) dans votre image Ink-

scape et ensuite de cliquer sur « Élément SVG sélectionné », ce qui remplit la boîte Image Source avec l'identifiant de l'élément. De cette façon, il devrait être possible d'incorporer n'importe quel autre élément SVG dans votre chaîne de filtrage... sauf que ça ne fonctionne pas. Cela a l'air de marcher dans la 0.48, puisqu'une version pixelisée de votre élément est incorporée et étirée pour remplir la boîte englobante, mais, dans la 0.91, même cette possibilité réduite a disparu.

Vous avez donc la primitive Image - un filtre qui promet beaucoup, mais en donne si peu. Les parties utiles qui fonctionnent dans la 0.48 sont hors d'usage dans la 0.91, alors que les parties utilisables de la 0.91 vous obligent à plonger dans l'éditeur XML. Dans le même temps, la triste interface utilisateur baisse les bras, se gausant de vos efforts pour réaliser quelque chose d'aussi audacieux que le réglage de la position de votre image dans la boîte englobante. Espérons que l'interface utilisateur sera plus étoffée dans une future publication, et que la possibilité d'utiliser les éléments SVG fera un retour bienvenu.

Ceci conclut notre tour des primitives de remplissage d'Inkscape. Cependant, la spécificité SVG en a une autre : Tile (Tuile), qui vous permet d'inclure

la sortie d'une autre primitive qui sera répétée (tuilée) sur la totalité de la région du filtre. Pour qu'elle marche, la primitive qui rentre doit avoir une région de filtre plus petite que celle où elle sera tuilée ; mais, comme Inkscape utilise une région de filtre unique pour la totalité de la chaîne de filtrage, même si cette primitive devait être implémentée, elle n'aurait aucun effet visible.

Ça ne paraît pas très sympa de vous avoir fait perdre votre temps à lire la présentation d'un filtre aussi pauvrement implémenté et d'un autre pas du tout implémenté ; aussi, je finirai cet article par une autre primitive utile de la boîte à outils : Morphologie.

**MORPHOLOGY:  
EROSION**

**MORPHOLOGY:  
EROSION**

**MORPHOLOGY:  
DILATION**

**MORPHOLOGY:  
DILATION**

En dépit de son joli nom, c'est un filtre très simple : tout ce qu'il fait est de rendre les choses plus épaisses ou plus fines. Et il le fait avec le minimum de chichis : il n'y a qu'une liste déroulante, pour sélectionner « Contracter » (rendre les choses plus fines) ou « Dilater » (les rendre plus épaisses) et une paire de réglettes « Rayon », liées en option, pour ajuster la quantité de finesse ou d'épaisseur appliquée. Regardons ce filtre en action ; dans chaque cas, le premier objet texte est non filtré, et le second est filtré comme indiqué, avec un rayon de 2,5.

Ces filtres sont particulièrement utiles quand ils sont utilisés avec la primitive Composite, souvent en mode « In » ou « Out ». Dans l'exemple suivant, j'ai utilisé un filtre Remplis-

**Fake 3D  
Effect**

**Fake 3D  
Effect**

sage pour créer un remplissage blanc translucide, puis j'ai utilisé une Composite « In » pour l'ajuster à la taille de mon texte contracté. Après un peu de Flou gaussien et d'Offset, vous obtenez un filtre qui donne une apparence 3D à votre texte.

« Out » peut bien fonctionner avec la dilatation, pour creuser le centre de l'image dilatée. Comme exemple simple, considérez une primitive Morphologie qui dilate la source, puis une Composite « Out » qui ne garde que les parties de cette image qui dépassent de l'objet source d'origine. Il ne reste que le contour de votre objet, dont l'intérieur est transparent.

Maintenant, plutôt que d'enfoncer l'objet source, que se passe-t-il si vous enfoncez une autre version dilatée, de sorte que vous enlevez une petite excroissance au cœur d'une plus grosse dilatation ? Fusionnez avec l'objet original et vous avez un contour qui entoure l'original, à une distance réglée par la plus petite dilatation, avec une épaisseur égale à la différence entre les dilatations intérieure et extérieure (vous devrez peut-être augmenter la taille de la région du filtre pour éviter que le résultat soit tronqué).

Enfin, que se passe-t-il si vous prenez l'idée précédente et la complétez

un peu plus ? Vous pouvez avoir plusieurs contours, à des distances différentes de l'objet d'origine, que vous fusionnez ensemble pour terminer. Plus vous ajoutez de contours, plus les choses deviennent compliquées, car vous jonglez avec une paire de primitives Morphologie et une Composite

**Outline Effect**

Outline Effect

Outline Effect

Outline Effect

pour chaque couche de l'oignon ; mais, théoriquement, il est possible d'en rajouter autant que vous voulez, tant que vous ne vous perdez pas en route.

**Multi-Coloured Outline Effect**

Et que dire de cette version, où j'ai utilisé aussi des primitives Matrice des couleurs en mode Décalage de teinte de façon à attribuer une couleur à chaque contour (bas de la colonne précédente).

Il est bon de se rappeler que les filtres sont des opérations bitmap qui ont lieu pendant la phase de rendu. Bien que vous puissiez penser que la primitive Morphologie rétrécit ou dilate votre image, ce n'est pas réalisé



**Erode**



**Dilate**

dans un sens vectoriel, mais plutôt en ajoutant ou en retirant des pixels dans une version bitmap de votre objet. En gardant cela en tête, il paraît possible d'appliquer la primitive à des images bitmap importées via la primitive Image. Ceci vous permet de cacher les détails précis d'une image en les dégradant un peu, ou en les masquant par une dilatation des zones adjacentes, sans introduire la sorte d'adoucissement que vous attendriez en floutant les images. Dans les deux cas, *Mona* se sort bien mal d'un tel essai !

## CRÉDITS IMAGES

*La Joconde* (alias *Mona Lisa*) par Léonard de Vinci [http://en.wikipedia.org/wiki/File:Mona\\_Lisa,\\_by\\_Leonardo\\_da\\_Vinci,\\_from\\_C2RMF\\_retouched.jpg](http://en.wikipedia.org/wiki/File:Mona_Lisa,_by_Leonardo_da_Vinci,_from_C2RMF_retouched.jpg)



Mark a utilisé Inkscape pour créer trois bandes dessinées, *The Greys*, *Monsters, Inked* et *Elvie*, qui peuvent toutes être trouvées à <http://www.peppertop.com/>





Une caractéristique principale des filtres SVG est leur dynamique. Les calculs pour produire la sortie ne sont pas calculés une fois puis stockés dans l'image, comme c'est souvent le cas avec les filtres des éditeurs bitmap. À la place, ils sont calculés chaque fois que vous zoomez, découpez, tournez vos objets ou modifiez votre dessin. Ceci vous donne la flexibilité de modifier vos paramètres de filtrage à tout moment, mais ces calculs affectent la vitesse de rendu d'Inkscape. Aussi, maintenant que vous commencez (je l'espère) à créer de plus en plus de filtres complexes, je vais commencer cet article en analysant quelques moyens de réduire ce ralentissement.

Quand vous êtes confronté à un programme qui ralentit à cause de trop nombreux calculs, il y a deux approches qui peuvent être utilisées pour minimiser le problème : réduire le nombre des calculs ou trouver une façon de les accélérer. Vous savez maintenant que, à la base, les filtres sont appliqués par pixel, juste au moment du rendu de l'objet, effectuer un zoom arrière est une manière de réduire les calculs. Un objet vu en vision serrée,

qui fait 10 pixels par 10 sur l'écran, occupe une surface de 100 pixels. Même pour le plus simple des filtres théoriques, ça veut dire 100 calculs, mais en principe, ça veut dire beaucoup plus, car, au strict minimum, les calculs pour les canaux rouge, vert, bleu et alpha devront probablement être distincts. Augmentez le champ pour que l'objet remplisse 20 pixels par 20 - ce qui serait couramment considéré comme « deux fois plus gros » - et la surface est quatre fois plus grande, soit 400 pixels et, par conséquent, 400 calculs par canal. Augmentez le champ de vision jusqu'à ce que votre petit objet remplisse presque toute la surface de votre moniteur HD et il y a beaucoup de calculs à faire !

Tout comme éviter des vues larges, vous pouvez réduire le nombre de pixels à recalculer en redimensionnant simplement votre fenêtre Inkscape. Doit-elle vraiment être dimensionnée à la taille de votre moniteur à écran large ? Essayez de réduire la taille du canevas dans une forme plus carrée au centre de votre écran, en laissant flotter les fenêtres de dialogue sur les côtés.

Parfois, vous n'avez pas vraiment besoin de voir la version filtrée d'un objet, si vous voulez juste agrandir le champ pour ajuster sa forme. Dans de tels cas, il y a l'option Afficher > Mode d'affichage > Sans filtre. Il y a aussi une option pour ne voir que le contour des objets, ce qui peut être utile pour trouver des objets que vous auriez perdus dans la multitude des façons de rendre les objets invisibles, mais qui n'offre réellement rien de plus pour ce qui est de la lenteur des filtres. Je le mentionne tout simplement parce qu'il y a aussi une option Alternier qui tourne sur les trois modes : si vous ne faites qu'une chose aujourd'hui, apprenez son raccourci clavier (CTRL-5 par défaut, où « 5 » est la touche du pavé numérique). Ce que ceci a de génial est que vous n'avez pas besoin de changer de mode avant de zoomer : si vous agrandissez l'image et que l'affichage est trop lent, appuyez juste sur CTRL-5 pour changer de mode, abandonnant ainsi le recalcul en cours.

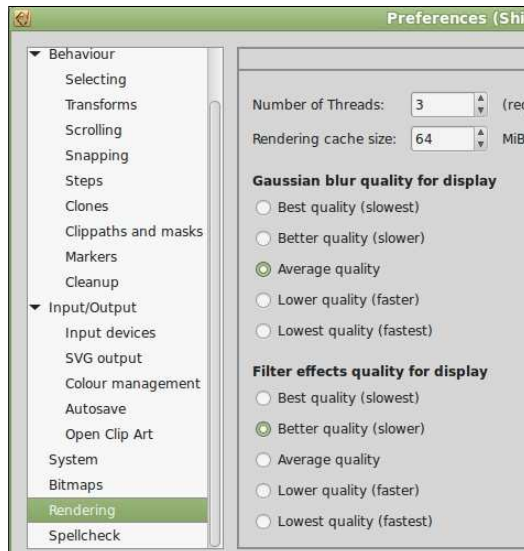
Que se passe-t-il quand vous avez fini d'ajuster un objet filtré, au moins à ce moment précis ? Si vous n'en avez pas besoin pour travailler sur les autres parties de votre dessin, c'est mieux

de le placer sur son propre calque ou sous-calque. Masquez ce calque, et Inkscape n'a plus rien à recalculer. Si vous voulez néanmoins toujours le voir, vous pouvez faire une copie bitmap de l'objet filtré avant de déplacer l'original sur un autre calque. Sélectionnez votre objet et utilisez Édition > Faire une copie bitmap (ou appuyez sur ALT-B) : Inkscape rendra un bitmap de votre objet, avec l'application des filtres, signifiant (une fois l'original masqué) qu'un recalcul des filtres n'est plus nécessaire pendant que vous travaillez sur votre document. La résolution de la copie bitmap est réglée dans les préférences d'Inkscape, la création sera plus rapide avec des valeurs basses, mais la précision sera moins bonne en agrandissant. Toutefois, cela n'a habituellement pas d'importance parce que le bitmap est présent généralement comme référence de position ou de couleur, plutôt que comme une représentation en haute résolution de votre objet.

Ces méthodes réduisent la quantité des calculs qui doivent être réalisés, mais il y a aussi des façons d'accélérer la performance du filtre, même quand vous avez besoin de



garder visibles les objets filtrés originaux. Dans les préférences d'Inkscape (Fichier > Préférences d'Inkscape dans la 0.48, Édition > Préférences dans la 0.91), il y a un panneau pour ajuster le rendu des filtres, appelé « Filtres » dans la 0.48 et « Rendu » dans la 0.91.



Dans ce panneau, vous pouvez régler le nombre de « threads » (tâches en parallèle) qu'Inkscape utilise pour le rendu de filtres de flou gaussien (0.48) ou des filtres en général (0.91). Si votre ordinateur a un processeur multi-cœurs ou « hyper-threading » (technologie multi-tâches), l'augmentation de la valeur peut accélérer le rendu des filtres. On recommande habituellement de le régler au nombre de cœurs moins 1, ce qui, en théorie, permet l'utilisation d'un seul cœur

pour le processus principal, tout en servant des cœurs restants pour le rendu des filtres. En pratique, il y a tout le système d'exploitation entre vous et vos cœurs ; aussi, bien que ce soit une indication utile, il n'y a pas de garantie que votre OS distribue les « threads » aussi nettement.

En plus, dans la 0.91, vous pouvez aussi réserver un peu de mémoire dans laquelle mettre en cache les résultats de vos calculs de filtres. Ceci aurait un effet sur des actions comme le déplacement dans la fenêtre - où un résultat de filtrage déjà calculé est déplacé pour être vu ou caché - mais il aura probablement moins d'effet si vous agrandissez ou réduisez l'échelle, car les filtres devront être recalculés de toute façon à chaque niveau. Néanmoins, si vous avez plein de RAM libre, ça peut valoir le coup d'en assigner un peu plus à cette option pour accélérer les choses là où c'est possible.

Enfin, deux réglages par boutons radio gouvernent la balance entre qualité et rapidité d'affichage. Les filtres peuvent être approximés par un rendu à faible résolution, offrant un rafraîchissement plus rapide de l'écran avec une moindre précision. Ces boutons vous permettent d'ajuster cet équilibre pour les filtres en général, mais aussi pour le flou gaussien en parti-

culier (car il est assez souvent la plus communément utilisée des primitives de filtrage). Notez que ces boutons radio n'affectent que l'affichage de votre image à l'écran, l'exportation en fichier PNG utilise toujours la plus haute qualité possible.

Abandonnant la performance, revenons aux filtres eux-mêmes. D'abord, une petite correction au précédent article : il semble finalement que la primitive Image dans la 0.91 vous permet malgré tout d'utiliser un élément SVG de votre image comme entrée. Le problème est que l'élément est inclus par rapport au coin en haut et à gauche de la page ; aussi, si vous essayez d'inclure quelque chose qui est situé loin de cet angle, il y a de bonnes chances que vous voyiez une zone blanche insérée dans votre filtre (ce qui m'avait fait penser qu'il ne fonctionnait pas du tout). Il y a deux solutions possibles à cela : dessinez votre élément SVG inclus en haut à gauche de la page (vous pouvez le placer sur un calque masqué si vous ne voulez pas le rendre visible à cet endroit dans l'image finale), ou augmentez la taille de la région de votre filtre jusqu'à ce que l'élément inclus soit visible, puis utilisez la primitive Décalage pour le déplacer au bon endroit. Aucune des deux n'est bien fameuse, à mon avis, mais, entre les deux, j'ai tendance à préférer le

placement de l'élément inclus (ou un de ses clones) dans le coin en haut et à gauche, sur un calque masqué, car l'autre solution entraîne des calculs sur une plus grande région de filtrage, et ainsi ralentit le rendu.

Un autre problème avec cette fonctionnalité dans la 0.91 apparaît si vous essayez d'utiliser le même objet à la fois comme cible de la chaîne de filtres et comme entrée de la primitive Image. C'est assez facile de le faire par erreur, car le défaut de l'interface utilisateur des filtres d'Inkscape augmente la probabilité que vous perdiez la trace de ce qui est sélectionné et pourquoi, mais le résultat est un plantage instantané d'Inkscape, sans avertissement et sans fichier de sauvegarde. Si vous prévoyez d'utiliser des objets SVG comme entrées de la primitive Image dans la 0.91, il vaut mieux enregistrer votre fichier juste avant d'ajouter le lien.

Utiliser la primitive Image en conjonction avec le filtre Carte de déplacement est une bonne idée. Celui-ci remplace individuellement chaque pixel de sortie par un autre pris ailleurs dans votre image ; ainsi des volutes, vagues et distorsions peuvent être créées. Il s'appuie sur deux entrées : la première est l'image que vous voulez distordre, tandis que la seconde

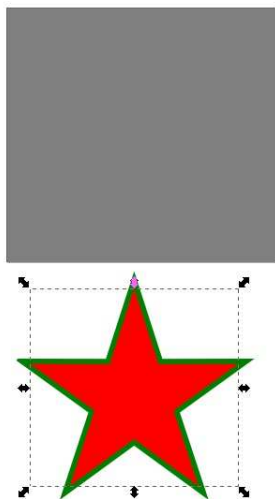
est une autre image qui agit comme une « carte » pour dire au filtre où trouver chaque pixel de sortie. Le processus est vraiment très simple quand on le considère sur une base par pixel, mais devient rapidement plutôt complexe si vous essayez de créer une carte de déplacement pour réaliser une distorsion spécifique.

Pour commencer à comprendre cette primitive, débutons avec la plus classique des chaînes :



Comme vous le voyez, la première entrée de carte de déplacement est notre source image, alors que la seconde vient de la primitive Image. En pratique, l'image n'est qu'un rectangle gris à 50 % incorporé comme élément SVG (et positionné dans le coin gauche en haut de la page, de façon à ce qu'il fonctionne dans la 0.91). Il y a

aussi deux étoiles dans l'image : le filtre est appliqué à la rouge, alors que la verte est là tout simplement comme référence pour que vous voyiez l'effet plus clairement. Les paramètres de l'effet sont une longueur/courbure de 10, avec les canaux rouge et vert utilisés comme sources respectives des déplacements X et Y - ceci deviendra plus clair sous peu.



Le résultat du filtre est... absolument rien ! Pour comprendre pourquoi, considérons un seul pixel de notre image de sortie. Ce pixel vient de quelque part dans notre source image, et la nature précise du « quelque part » est définie par la carte de déplacement (la seconde image d'entrée). Chaque pixel dans la carte de déplacement est composé d'une combinaison de quatre valeurs (rouge, bleu, vert et alpha) et les réglages dans le

dialogue du filtre vous laissent choisir laquelle de ces valeurs devrait être utilisée pour le décalage X, et laquelle pour le décalage Y. À partir de là, Inkscape exécute les étapes suivantes pour déterminer quelle sera la couleur du pixel de sortie :

- 1) Chercher la couleur du pixel équivalent dans la carte de déplacement.
- 2) Extraire les décalages X et Y des composantes de couleur réglées dans le filtre.
- 3) Diviser les décalages par 255 pour les normaliser dans une plage de 0 à 1.
- 4) Soustraire 0,5 des décalages pour les déplacer dans la plage -0,5 à 0,5.
- 5) Multiplier les décalages par la valeur de longueur/courbure définie dans le filtre.
- 6) Ajouter les valeurs des décalages aux coordonnées X et Y du pixel pour obtenir une nouvelle paire de coordonnées.
- 7) Le pixel de sortie sera réglé à la couleur du pixel de l'image d'entrée qui est situé aux nouvelles coordonnées, ou à une couleur interpolée à partir de celles des pixels voisins si les coordonnées n'indiquent pas un seul pixel.

Gardez en tête que notre carte n'est que du gris à 50 %, avec les valeurs RGB à 127, 127, 127. Si vous suivez les étapes ci-dessus, vous trou-

verez un décalage de -0,02 pixels pour X comme pour Y, assez proches de zéro pour signifier en fait que le pixel de sortie prend la même position que le pixel d'entrée. Appliquez cela à tous les pixels du filtre et il est clair que notre sortie ressemblera exactement à notre entrée.

Le changement du rectangle en noir (0, 0, 0) modifie quelque peu le calcul. Maintenant, le décalage devient -5, -5 ; aussi, notre pixel de sortie prend la couleur du pixel situé un peu au-dessus et un peu à gauche de l'image d'origine. Ainsi, toute l'image paraît avoir bougé vers le bas et à droite.



La modification du rectangle en blanc (255, 255, 255) a l'effet inverse : l'image semble bouger vers le haut et à gauche. Parce que nous avons spécifié le rouge et le vert pour les déplacements X et Y, un remplissage tout en rouge (255, 0, 0) produit des valeurs de déplacement différentes pour les deux coordonnées, déplaçant l'image en bas à gauche ; un pur vert (0, 255, 0) la déplace en haut à droite. Dans tous les cas, la valeur de la compo-



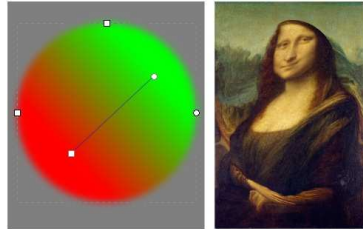
sante bleue (ou, bien sûr, de la composante alpha) ne fait aucune différence. Un cyan pur (0, 255, 255) à le même effet que le vert pur, car nous avons configuré le filtre pour ne réagir qu'aux composantes rouge et verte.

L'utilisation d'une couleur unie pour la carte de déplacement n'est qu'un pâle remplacement de la primitive Décalage. Là où elle prend tout son sens, c'est quand la carte de déplacement contient des couleurs variées de façon à utiliser des décalages différents pour chaque pixel. Nous savons qu'un remplissage noir tire le pixel vers le haut à gauche, et qu'un blanc l'entraîne en bas à droite ; qu'arrive-t-il quand on utilise une image contenant du noir et du blanc ? Faisons un essai avec un groupe contenant une spirale noire sur un fond blanc et nous l'appliquons à quelque chose de plus complexe qu'une étoile rouge.



En ajoutant un peu de flou gaussien entre la primitive Image et la carte de déplacement, nous pouvons adoucir les bords pour donner un joli effet d'ondulations, son intensité étant

réglée par le paramètre longueur/courbure. Et pourquoi pas une variation du rouge au vert pour donner un effet du genre « fish eye » (grand angle, littéralement œil de poisson) ?



C'est tricher un peu, car l'utilisation des seuls rouge et vert n'étire l'image que dans deux directions. La superposition d'un cercle passant du blanc au transparent puis au noir sur un axe perpendiculaire donne un effet plus précis, mais commence à signaler le plus gros problème de la primitive Carte de déplacement : créer une image carte convenable pour l'effet que vous voulez n'est pas toujours simple ou évident. Mais il y a une façon assez simple et extrêmement utile de créer une carte : la primitive Turbulence.

Si vous avez besoin de vous rafraîchir la mémoire à propos de cette primitive, regardez la partie 51 de cette série. En bref, c'est une façon rapide de créer des zones remplies de couleurs pseudo-aléatoires qui, quand elles sont utilisées dans une carte de déplacement, tireront l'image d'une



façon ou d'une autre suivant le réglage des paramètres. Utilisez un réglage de bruit fractal à basse fréquence pour ajouter des distorsions grotesques à votre image. Augmentez un peu les valeurs pour produire le genre de distorsions de flou que vous pouvez voir à une fenêtre de salle de bain. Encore plus loin et vous obtiendrez un chef-d'œuvre du pointillisme aux pixels détachés. Déliez les fréquences horizontale et verticale et vous avez un drapeau flottant au vent ou des ondulations horizontales.

Mais prenez bien le temps de regarder les bords. Et quels bords ! Des ondulations légères, en passant par des éclaboussures d'encre, jusqu'à un vignettage flou. Imaginez à quoi de tels filtres peuvent ressembler sur des formes avec encore plus de côtés, comme les carrés, les étoiles ou du texte. Encore mieux, n'oubliez pas ; remontez vos manches, plongez dans l'éditeur d'Inkscape et créez vos propres filtres.

Crédits images *La Joconde* (alias *Mona Lisa*) par Léonard de Vinci. [http://en.wikipedia.org/wiki/File:Mona\\_Lisa,\\_by\\_Leonardo\\_da\\_Vinci,\\_from\\_C2RMF\\_retouched.jpg](http://en.wikipedia.org/wiki/File:Mona_Lisa,_by_Leonardo_da_Vinci,_from_C2RMF_retouched.jpg)



Mark a utilisé Inkscape pour créer trois bandes dessinées, *The Greys*, *Monsters*, *Inked* et *Elvie*, qui peuvent toutes être trouvées à <http://www.peppertop.com/>



Ce mois-ci, nous allons examiner la primitive de filtrage Matrice de convolution. La convolution est le terme mathématique pour le mécanisme d'application répétitive d'une fonction à la sortie variable d'une autre fonction. Dans le monde informatique, elle est habituellement utilisée avec des valeurs discrètes (c'est-à-dire séparées), plutôt qu'avec des valeurs continues, telles que vous pouvez en obtenir quand vous traitez de l'audio échantillonné, ou bien sûr, des pixels individuels d'une image. Aussi, en traitement numérique du signal, la convolution signifie généralement qu'on utilise une fonction pour représenter une série de valeurs dans une nouvelle série. En termes de filtres SVG, cela signifie représenter un ensemble de pixels dans un autre ensemble. La « fonction » est définie par l'utilisation d'une matrice de nombres, d'où le nom « Matrice de convolution » du filtre - bien que l'expression « représentation des valeurs des pixels par l'utilisation d'une matrice » l'aurait peut-être rendue un peu plus compréhensible au néophyte.

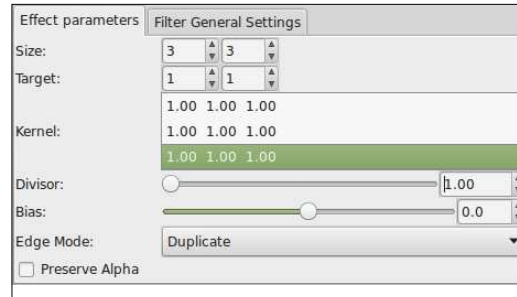
Regardons comment fonctionne une matrice de convolution en illustrant son effet sur une image simple

faite d'un petit ensemble de pixels. Pour cette démonstration, nous utiliserons des pixels noirs et blancs avec les valeurs 0 et 255 (les nombres intermédiaires étant des niveaux de gris). Dans un vrai filtre, il y a trois canaux de couleurs ; aussi notre image monocal en noir et blanc est seulement un modèle pour représenter la logique de calcul. La forme que nous utiliserons sera un simple carré de 9 pixels dans un carré plus grand de 25 pixels.

255	255	255	255	255
255	0	0	0	255
255	0	0	0	255
255	0	0	0	255
255	255	255	255	255

Notre première matrice sera un tableau 3x3, avec le nombre 1.00 dans chaque case, et la « cible » spécifiée comme étant la cellule centrale de la matrice. Voici à quoi ça ressemble dans l'interface utilisateur d'Inkscape (image en haut de la troisième colonne).

Le procédé de convolution lui-même consiste à prendre notre matrice et à



la positionner de sorte que la cellule cible de la matrice soit tour à tour positionnée au-dessus de chaque pixel de l'image d'entrée. Nous allons regarder le calcul qui a lieu pour le premier pixel noir de l'image d'entrée, celui avec un contour rouge. Les 9 pixels recouverts par la matrice sont tous multipliés par la valeur correspondante de la cellule de la matrice, puis additionnés ensemble. Le résultat est borné, de sorte qu'il ne peut pas excéder 255

255	255	255	255	255
255	1.00	1.00	1.00	255
255	0	0	0	255
255	1.00	1.00	1.00	255
255	0	0	0	255
255	0	0	0	255
255	255	255	255	255

ou passer sous 0, et il est utilisé ensuite comme valeur du pixel de sortie. Cette image peut clarifier un peu les choses : la zone verte représente la matrice 3x3, avec une représentation de la contribution de chaque pixel à la sortie.

La valeur du pixel de sortie est donc :

$$\begin{aligned}
 &(255 \times 1.00) + \\
 &(255 \times 1.00) + \\
 &(255 \times 1.00) + \\
 &(255 \times 1.00) + (0 \times 1.00) + \\
 &(0 \times 1.00) + (255 \times 1.00) + \\
 &(0 \times 1.00) + (0 \times 1.00)
 \end{aligned}$$

Pas besoin de grandes connaissances mathématiques pour comprendre que les cinq pixels blancs ont apporté une valeur de 255 à la sortie,

255	255	255	255	255
255	1.00	1.00	1.00	255
255	0	0	0	255
255	1.00	1.00	1.00	255
255	0	0	0	255
255	0	0	0	255
255	255	255	255	255



alors que les pixels noirs ne contribuent en rien. Aussi, la valeur utilisée pour le pixel de sortie est simplement  $255 \times 5 = 1\,275$ . Sauf que les valeurs de sortie sont limitées, de sorte que la vraie valeur de sortie est 255 seulement, cette matrice change le pixel noir en un pixel blanc.

Sur le pixel suivant, le résultat est identique. Cette fois-ci, seulement trois pixels blancs contribuent à la sortie ; mais la valeur est encore de 765, qui est bornée, et donc la sortie est à nouveau blanche.

En considérant les pixels noirs restants dans notre image, il paraît assez évident que tous ceux de l'extérieur vont virer au blanc. En fait, seul le pixel tout au centre reste noir. Ainsi, la sortie de cette matrice de convolution particulière est un simple pixel noir au centre d'un carré blanc.

Certains d'entre vous auront noté que j'ai commencé astucieusement sur un pixel qui n'est pas au bord de la zone de filtrage. Comment fait Inkscape pour calculer la valeur du pixel en haut à gauche, par exemple, étant donné que cinq des points couverts par la matrice n'existent simplement pas ? La réponse se trouve dans la liste déroulante « Mode bordure » des réglages du filtre : « Dupliquer » copie

les pixels du bord extérieur pour remplir toutes les valeurs manquantes ; « Retour à la ligne » utilise les pixels du côté opposé de l'image pour remplir les vides, comme si elle travaillait sur une version tuilée de l'image d'entrée ; « Aucun » met les valeurs des canaux des pixels manquants à zéro.

Ou, du moins, c'est ainsi que c'est censé fonctionner. D'après le manuel officiel Inkscape, ce paramètre est complètement ignoré par Inkscape, bien qu'il soit présent dans l'interface. La méthode pour calculer les pixels manquants n'est pas indiquée, et, comme le manuel officiel n'a pas été remis à jour pour la version 0.91, je ne sais pas si la situation a changé avec la version la plus récente. Aussi, nous ignorons cette question et considérerons qu'Inkscape remplit les pixels manquants ou les néglige lors du calcul, de sorte de ne pas trop nous en faire à ce propos.

Parce que les valeurs que nous avons choisies entraînent une troncature des résultats de calcul, notre filtre, tel qu'il est, crée simplement en sortie des pixels noirs et blancs. Avant la troncature, les résultats étaient 1 275 et 765, mais notre action suivante supprime toute différence de valeur parce qu'ils sont tous au-dessus de 255. En utilisant le contrôle Diviseur dans

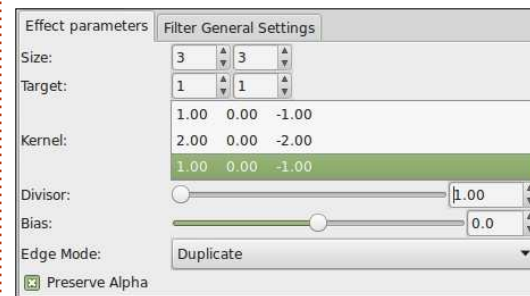
les paramètres du filtre, nous pouvons adapter la sortie des calculs avant troncature, nous permettant de restreindre les valeurs pour préserver ces différences. Une bonne règle empirique est de mettre le diviseur à la même valeur que la somme de chacun des nombres de la matrice. En le réglant à 9 dans notre exemple, les sorties 1 275 et 765 sont réduites à 142 ( $1\,275 \div 9$ ) et 85 ( $765 \div 9$ ), nous donnant ce résultat :

255	255	255	255	255
255	142	85	142	255
255	85	0	85	255
255	142	85	142	255
255	255	255	255	255

Ici, chaque pixel de sortie est la moyenne de neuf pixels de l'image d'entrée. Bien que cela puisse ne pas être clair dans ce petit exemple, le résultat est un simple floutage de l'image d'entrée. En réalité, il serait mieux d'utiliser une primitive Flou gaussien si vous voulez juste adoucir un peu votre image, mais c'était, bien sûr, juste une démonstration des mathématiques qui

se déroulent en arrière-plan de la Matrice de convolution.

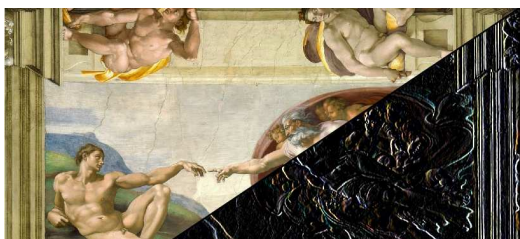
Maintenant, passons à des matrices plus intéressantes. J'arrête l'approche pixel par pixel et les explications mathématiques, c'est tout simplement une extension des exemples que je vous ai montré jusqu'à présent, mais avec des images plus grandes et plusieurs canaux de couleur. Nous utiliserons une image classique différente pour les présenter car *la Joconde*, très honnêtement, n'est pas intéressante pour l'application d'une matrice de convolution. Aussi, nous prendrons à la place *la Création d'Adam* de Michel-Ange, avec, pour chaque image, la version non filtrée en haut à gauche et la version filtrée en bas à droite. Nous commencerons par une matrice « Sobel » :



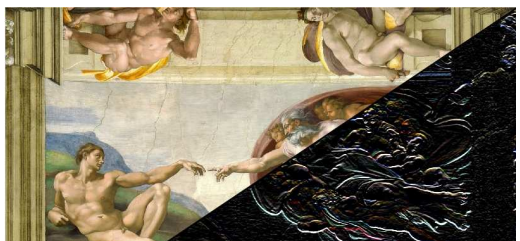
Un opérateur Sobel amplifie les différences entre les pixels voisins dans une direction ou une autre. Le résultat est essentiellement une configuration avec des couleurs lumineuses,



où la transition entre pixels est franche, et des couleurs sombres où il y a peu ou pas de différences entre les pixels adjacents. Par conséquent, dans la pratique, il agit comme un filtre de détection des contours, dans ce cas, soulignant les contours verticaux (notez particulièrement la moulure à droite de l'image).



La rotation des valeurs de la matrice de  $90^\circ$  (de telle sorte que la ligne du haut contienne 1, 2, 1 et celle du bas -1, -2, -1) la transforme en un filtre de détection des contours horizontaux. Dans ce cas, la moulure disparaît, mais les formes presque horizontales sont accentuées :

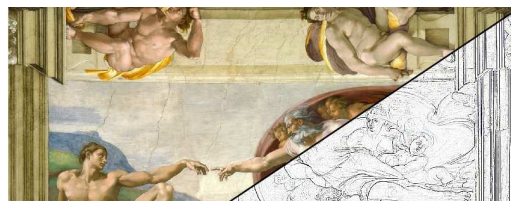


Une forme plus générale de détection des contours, qui souligne les lignes verticales et horizontales donnant une version « schématique » de l'image d'origine, peut être réalisée

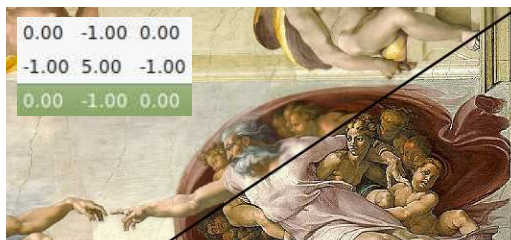
avec la matrice suivante :



Il est facile d'imaginer ceci, suivi par une primitive Matrice des couleurs, formant la base d'une chaîne de filtrage d'« esquisse au crayon », mais vous pouvez obtenir un résultat similaire en utilisant le paramètre Déviation dans les préférences de filtrage. Celui-ci vous permet d'ajouter un décalage fixe au résultat de chaque calcul et agit pour assombrir ou éclaircir l'image de sortie. Le réglage de ce paramètre à 1.0 avec le filtre précédent donne ce résultat :



Une variation de la détection des contours est l'amélioration des contours. Cette matrice amplifie les contours tout en permettant aux cou-



leurs d'origine de disparaître, donnant une apparence accentuée de l'image (bas de colonne précédente).

Voici une autre matrice qui assombrit certains contours tout en en



éclaircissant d'autres, donnant vie à une sorte d'embossage. Comme vous pouvez le voir, il y a une large palette d'effets qui peuvent être produits avec cette primitive, bien que déterminer quelles valeurs saisir pour obtenir une sortie particulière soit peu intuitif. Bien que l'approche par matrice permette beaucoup de possibilités, il n'y a en réalité que peu de matrices bien connues qui sont utilisées habituellement. Une recherche en ligne vous fournira quelques autres exemples, mais ils rejoignent tous les thèmes de base que je vous ai présentés ici.

Avant de conclure, il y a quelques contrôles supplémentaires dans l'interface du filtre, que je dois expliquer. Le paramètre Taille, comme vous pouvez vous y attendre, détermine la taille de la matrice. Je n'ai utilisé que des matrices 3x3 dans cet article, mais 5x5 est aussi une taille classique, et vous pou-

vez augmenter encore plus, pour prendre en compte une zone plus large autour de chaque pixel de la source. Simplement, gardez en tête que la taille de la matrice définit le nombre de pixels qui doivent être lus et calculés pour chaque pixel de sortie ; aussi, l'augmentation de ce paramètre peut rapidement imposer une charge de traitement beaucoup plus grande à Inkscape.

Ici, dans ces exemples, j'ai pris comme base que le centre de la matrice est positionné sur le pixel cible de chaque calcul. Il est possible de changer cela en utilisant le champ Cible de l'interface, où 0,0 placera comme cible la cellule en haut à gauche de la matrice. Tout ceci décale un peu la sortie ; aussi, il n'y a pas de raison d'en faire grand cas.

Enfin, la case à cocher Préserver l'opacité détermine si l'alpha du pixel d'origine est transféré sans changement à la sortie (coché), ou si le canal alpha est aussi traité par le processus de convolution (décoché). J'ai tendance à le garder coché, car ça fait un canal de moins à calculer pour Inkscape et je n'ai pas encore ressenti le besoin de convoluer le canal alpha.

## Crédits images

la Création d'Adam par Michel-Ange <https://fr.wikipedia.org>.

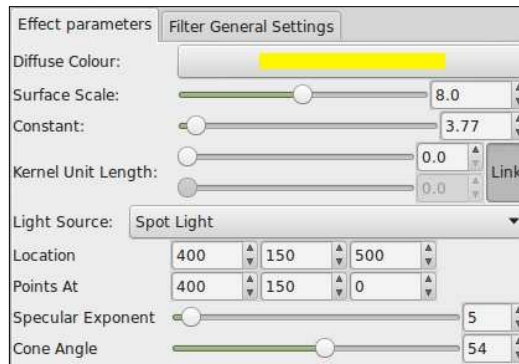




Ce mois-ci, nous examinerons les dernières primitives de filtrage disponibles dans Inkscape 0.48, Lumière diffuse et Lumière spéculaire. Elles sont utilisées pour simuler l'effet de la lumière brillante sur vos objets et constituent les deux tiers du modèle de réflexion Phong. La troisième partie, Lumière ambiante, fait référence à la lumière qui est présente en tous points d'une image plutôt qu'à celle venant d'une source de lumière particulière. Il n'y a pas besoin d'un filtre spécifique pour cette partie, car elle est formée par les couleurs de remplissage et de contour des objets de votre image.

Lumière diffuse fait référence à l'éclairage et l'ombrage généraux sur un objet, qui ne se modifient pas significativement quand vous déplacez votre point de vue. Lumière spéculaire, au contraire, fait référence aux points et reflets brillants qui tournent et se déplacent quand vous bougez. Regardez un objet brillant près de vous et déplacez votre tête pour voir la différence : notez que les éclats réfléchis par les bords et les coins bougent avec vous et que les ombres et reflets diffus sur le corps principal de l'objet restent largement inchangés.

Pour commencer, créez un objet ou un groupe auquel appliquer le filtre et ajoutez ensuite la primitive Lumière diffuse de la façon habituelle. Il y a quelques paramètres modifiables, mais pour la plupart, il s'agit de déplacer les réglettes par tâtonnements de manière à obtenir le résultat que vous recherchez.



Le premier paramètre à choisir est la couleur de la lumière. Celle-ci a un énorme effet sur la sortie du filtre, car l'effet d'éclairage remplace complètement la couleur originale de vos objets, plutôt que de se mélanger avec les teintes sous-jacentes. Dans l'exemple qui suit, tous les objets texte sont bleu sarcelle (une couleur bleu-vert), mais la couleur utilisée dans les filtres est jaune. Notez qu'aucun bleu sarcelle n'apparaît dans les images de sortie.

En pratique, seul le canal alpha de l'image d'entrée est utilisé par cette primitive ; aussi, le fait d'être connecté à la Source image ou à l'Opacité de la source n'a aucune importance : le résultat est le même. Le canal alpha est utilisé comme une « carte en relief » pour déterminer la position de chaque pixel le long de l'axe z ; les zones plus opaques ressortent du fond. Les réglettes Relief de surface et Constante peuvent être utilisées pour mettre à l'échelle et décaler les valeurs alpha de façon à modifier la profondeur apparente de l'objet.

Le paramètre Unité de longueur du kernel peut être largement ignoré. Il n'est pas utilisé par Inkscape, mais peut avoir de l'effet dans d'autres visionneuses SVG, où il sert à définir la taille de la grille des pixels utilisée pour les calculs de filtrage. Je le laisse habituellement à zéro.

Enfin, c'est le moment de choisir le type de source lumineuse : distante, ponctuelle ou spot. La première indique une source de lumière qui est à une distance infinie de l'objet, de telle sorte que tous les rayons lumineux incidents sont parallèles. Le paramètre

Azimut règle la localisation de la source lumineuse avec un angle. 0° la place à droite de votre objet ; en augmentant les valeurs, elle tourne autour de l'image, dans le sens horaire, jusqu'à 360° où elle est de retour sur la droite. Tirez la réglette pour voir l'effet en temps réel. Le paramètre Élévation règle l'angle par rapport au plan du dessin : imaginez une lampe posée dans le plan de votre écran d'ordinateur (peu de relief, des ombres sombres) ; au fur et à mesure que vous déplacez la réglette vers 90°, la lumière jaillit vers vous hors de l'écran, jusqu'à ce qu'elle soit pile au-dessus de vos objets. Continuez vers 180° et elle parcourt le même arc jusqu'à être au niveau du moniteur de l'autre côté de votre image ; n'importe quelle valeur supérieure continue à déplacer l'éclairage en demi-cercle derrière le moniteur et ce n'est pas vraiment utile.

La spécification de deux valeurs angulaires comme cela définit une position sphérique dans l'espace à trois dimensions. Si vous avez déjà regardé un épisode de Star Trek où un membre de l'équipage annonce leur déplacement comme « 249 mark 48 », c'est ce qu'ils font : simplement définir un azi-

mut et une élévation pour décrire la direction prise par le vaisseau. Ça m'amuse toujours de les voir capables de définir ces valeurs au degré près, mais il est vrai que je n'ai pas bénéficié d'un cours de cartographie inter-stellaire à la Starfleet Academy !

Avec deux valeurs polaires capables de définir une position sphérique, il suffit d'un troisième paramètre, la distance, pour spécifier un point dans l'espace. En sélectionnant l'option Lumière ponctuelle, vous pouvez vous attendre à voir les deux mêmes réglettes, associées à une troisième. Mais le groupe de travail SVG a décidé que la définition d'un point particulier en 3D devait être faite en utilisant les coordonnées cartésiennes ; aussi, à la place, vous avez trois champs anonymes avec une étiquette unique « Localisation », représentant la position du point par les coordonnées x, y et z. Il n'y a aucun moyen de choisir une localisation x, y sur le canevas et les valeurs sont en fonction du système de coordonnées de l'objet éclairé (qui n'est pas nécessairement le même que celui du dessin principal). Ainsi, une fois encore, c'est une cause de tâtonnements.

Alors que la Lumière distante, à une distance infinie de la scène, projette une illumination uniforme, la Lumière ponctuelle est bien plus nuancée.

Elle illumine plus les zones près de la source lumineuse que celles à distance, entraînant des nuances dans la couleur finale.

L'option Lumière spot est encore plus précise dans son effet. Elle requiert deux jeux de coordonnées : l'un pour spécifier la position de la lumière et l'autre pour définir la direction vers laquelle elle pointe (ce qui est en fait réalisé en spécifiant le point de l'espace qu'elle vise). La lumière est projetée dans un cône, de la source vers la cible, avec deux réglettes additionnelles pour ajuster les caractéristiques de ce cône : l'Exposant spéculaire définit la concentration du faisceau, tandis que l'Angle du cône définit la forme de celui-ci. Le cône est à bord vif ; tout point en dehors n'est pas du tout illuminé. Vous aurez donc besoin d'étapes de filtrage supplémentaires si vous voulez une bordure plus progressive.

Cet exemple montre l'utilisation des trois types d'éclairage sur des objets texte, qui ont vraiment tous le bleu comme couleur de base !

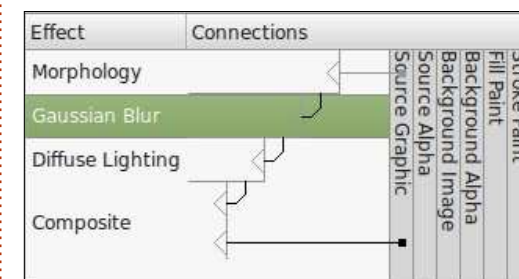
Vous noterez comme ils ont tous l'air « plats ». Parce que l'embossage est créé à partir du canal alpha de l'image d'entrée, et que les valeurs alpha de notre image d'entrée sont uniquement 0 et 255, il n'y aucune



possibilité pour des hauteurs variables. Si vous voulez adoucir le bord de votre éclairage, vous aurez besoin d'introduire un peu de variété dans le canal alpha. La façon la plus facile de le faire est d'utiliser une primitive Flou gaussien sur l'image d'entrée.

Le simple ajout d'un flou tendra à disperser le bord de votre texte à l'intérieur comme à l'extérieur (seconde image de l'exemple). Pour un effet plus prononcé, il est souvent pas mal d'utiliser un filtre Morphologie qui érode l'image d'entrée avant de la flouter. En affinant d'abord vos objets, l'extension complète du flou peut être contenue dans les limites des formes d'origine (troisième image). Si, alors, vous ajoutez un filtre Composite, réglé sur « In », à la sortie de la primitive Éclairage, vous pouvez rogner le résultat pour que vous ayez quelque chose de

plus proche du texte arrondi que, sans doute, vous souhaitiez (4<sup>e</sup> image).



Et malgré tout, nous conservons la couleur jaune venant du filtre Éclairage. C'est ici que l'option « Arithmétique » du filtre Composite est utile (relisez la partie 50 si vous avez besoin de rafraîchissement sur cette primitive). La sortie du filtre Lumière diffuse est destinée à être multipliée à l'image source pour superposer l'effet lumineux sur les objets sous-jacents ; mais, plutôt que de fournir un bon raccourci évident pour cette opération, l'interface utilisateur d'Inkscape offre juste les paramètres du modèle SVG



sous-jacent. Pour chaque canal de chaque pixel, l'opérateur Arithmétique réalise le calcul suivant :

$$\text{résultat} = (K1 \times i1 \times i2) + (K2 \times i1) + (K3 \times i2) + K4$$

où K1-4 sont les constantes réglées dans l'interface et où i1 et i2 représentent les valeurs de deux images d'entrée. En réglant K1 à 1.0 et toutes les autres constantes à 0, l'équation se simplifie en :

$$\text{résultat} = i1 \times i2$$

Autrement dit, une simple multiplication des valeurs d'entrée, ce qui est exactement ce que nous voulons. En changeant l'opérateur « In » dans la chaîne de filtres précédente par « Arithmétique », et en ajustant les constantes à 1, 0, 0, 0, la sortie est dans un ton



vert, résultat d'une lumière jaune brillant sur un objet bleu sarcelle.

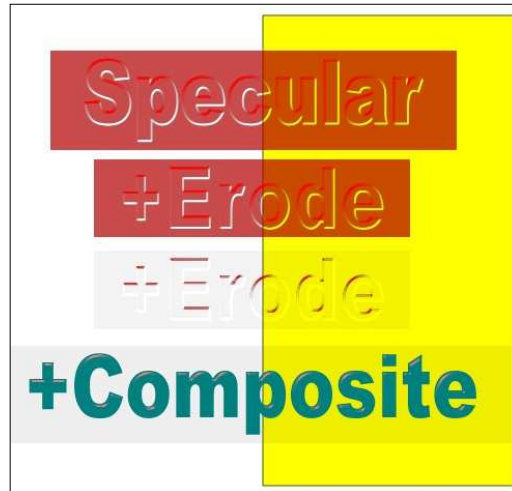
Maintenant, nous avons un objet illuminé dont la couleur de base a un effet sur la sortie. C'est évidemment beaucoup plus souple qu'un simple opérateur « In », qui nous obligerait à changer la lumière d'éclairage dans le

filtre lui-même chaque fois que nous voulons modifier le résultat. Si vous êtes inquiet de la perte de l'effet de découpe de l'opérateur « In », rassurez-vous : l'opérateur de multiplication s'applique aussi au canal alpha, de telle sorte que toutes les zones de l'image source où alpha=0 auront aussi des pixels transparents dans l'image de sortie.

En passant au filtre Lumière spéculaire, le contenu de l'interface utilisateur paraît très similaire. Il y a un paramètre supplémentaire, mais, autrement, c'est identique à la primitive Lumière diffuse. Le paramètre en plus est « Exposant », qui, d'après la spécification SVG et les conseils d'Inkscape, est utilisé pour rendre plus brillante la lumière spéculaire.

Contrairement à la Lumière diffuse, ce filtre produit une image avec des valeurs alpha variées. Faites attention, car des réflexions semblent brillantes pourraient n'être réellement que le fond blanc qui transparait ! Dans l'image suivante, vous pouvez voir cet effet très clairement sur les premier et second exemples, où les réflexions blanches « brillantes », à gauche du texte filtré, sont transformées en trous dans le canal alpha une fois qu'un fond jaune a été ajouté. Notez que j'ai utilisé une Lumière ponctuelle rouge dans ces

exemples, mais toujours avec le texte bleu sarcelle comme objet d'origine.



Les quatre images ci-dessus montrent l'effet du filtre Lumière spéculaire sur un texte uni, puis sur la version érodée du même. Cette fois, je n'ai pas ajouté un filtre Flou gaussien, car je voulais que les réflexions spéculaires soient franches et claires. En remontant la valeur d'Exposant dans la troisième image, la sortie s'approche de la seule vision des éclats, qui peuvent alors être ajoutés à nouveau à l'image source d'origine en utilisant une autre primitive Composite (quatrième image).

Cette fois-ci, le mode « Arithmétique » est utilisé à nouveau, mais les valeurs sont 0, 1, 1, 0, ce qui a pour effet de réduire l'équation à :

$$\text{résultat} = i1 + i2$$

Par conséquent, cette primitive ajoute les réflexions à l'image d'origine, ce qui est l'approche recommandée par la spécification SVG. Notez, cependant, qu'une légère opacité du fond est apparue ; aussi, vous pourriez vouloir ajouter un autre filtre Composite, réglé sur « In », pour assurer que le résultat est détourné à la taille des objets d'origine.

Enfin, il est temps de combiner les deux filtres d'éclairage pour produire une image parfaitement éclairée, avec les deux lumières diffuse et spéculaire. Une fois encore, le texte originel est bleu sarcelle et la lumière diffuse

## Full Lighting

jaune lui donne une apparence verte, mais vous pouvez aussi voir apparaître les reflets scintillants de la source lumineuse rouge du filtre spéculaire.

La chaîne de filtrage complète pour cet effet n'est pas trop compliquée si vous la faites étape par étape. En premier, la primitive Morphologie érode un peu le texte de l'image source et sa sortie va directement dans l'entrée de la primitive Lumière spéculaire pour obtenir des reflets nets et rouges. La sortie de Morphologie entre aussi dans un Flou gaussien pour adoucir l'image

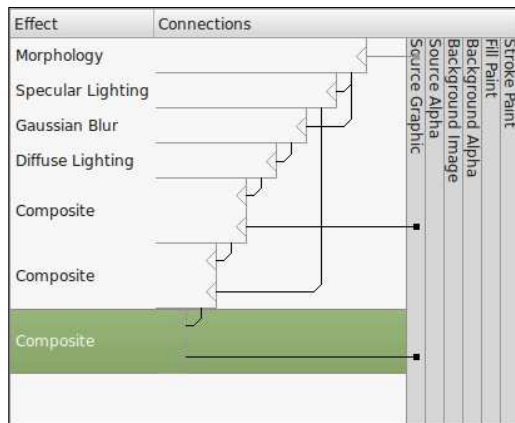
# TUTORIEL - INKSCAPE

avant qu'elle ne soit utilisée dans une primitive Lumière diffuse.

De là, il s'agit juste de tout combiner ensemble : le premier filtre Composite (mode « Arithmétique » : 1, 0, 0, 0) multiplie la source image avec la sortie d'Éclairage diffus. Le second Composite (mode « Arithmétique » : 0, 1, 1, 0) ajoute les accentuations de l'Éclairage spéculaire. Bien que le résultat soit presque parfait, le fond est légèrement visible, nonobstant une valeur alpha faible. Un troisième filtre Composite (mode « In ») arrange le tout un petit peu.

Bien que ce ne soit pas un vrai rayonnement ou un modelage 3D, les effets d'éclairage de SVG peuvent être utiles pour ajouter une fausse épaisseur à vos objets. Nul besoin d'être aussi évident que le texte 3D présenté ici : un simple reflet peut juste transformer une texture particulière-

ment fade en quelque chose de beaucoup plus intéressant, ou faire ressortir vos objets du fond. Comme d'habitude, la meilleure solution pour découvrir ce qui peut être fait avec eux est de les expérimenter.



**Mark** a utilisé Inkscape pour créer trois bandes dessinées, *The Greys*, *Monsters*, *Inked* et *Elvie*, qui peuvent toutes être trouvées à : <http://www.peppertop.com/>

## ÉDITIONS SPÉCIALES INKSCAPE (1 À 6) :



<http://www.fullcirclemag.fr/?download/303>



<http://www.fullcirclemag.fr/?download/398>



<http://www.fullcirclemag.fr/?download/477>



<http://www.fullcirclemag.fr/?download/483>



<http://www.fullcirclemag.fr/?download/485>



<http://www.fullcirclemag.fr/?download/489>

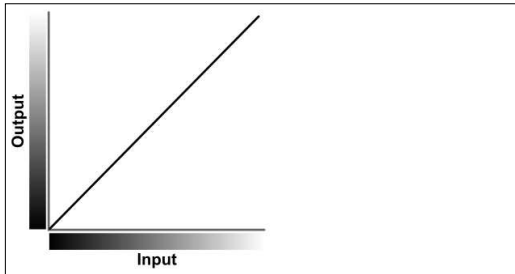




Il reste une dernière primitive de filtrage à visiter dans cette série, que j'ai gardée jusqu'à maintenant simplement parce que c'est un nouvel ajout dans la 0.91 ; elle est donc indisponible pour les utilisateurs qui se servent encore de la version 0.48. Le filtre s'appelle Transfert de composantes et son but est d'utiliser une fonction (appelée « fonction de transfert ») pour ajuster la distribution des valeurs dans chaque canal de couleur (ou « composante »). Il vous permet d'ajuster la luminosité ou le contraste, ou de régler des seuils forts pour des effets de postérisation. Comme d'habitude, je commencerai par m'intéresser aux opérations de filtrage sur un seul canal, puis vous pourrez extrapoler à partir de là pour voir comment il se comporte avec trois canaux plus l'alpha.

Une couleur unique d'un seul pixel est représentée par un nombre de 0 (pas de couleur) à 255 (complètement saturée). La distribution de ces valeurs est linéaire - augmentant le long d'une ligne droite - et les valeurs par défaut de la primitive Transfert de composantes laissent cette ligne inchangée. Une valeur de 0 dans le filtre entraîne une sortie à 0. 136 donne

une sortie à 136. Et ainsi de suite. Ceci peut être représenté dans un graphique, où la valeur du canal entrant dans le filtre est présenté sur l'axe x, et la valeur qui sort du filtre est visible sur l'axe y.



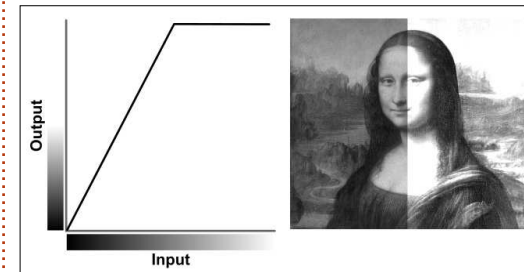
En pratique, cette primitive dispose les valeurs d'entrée entre 0 et 1, plutôt qu'entre 0 et 255, mais le résultat est le même : avec les valeurs par défaut du filtre (« Identité »), chaque canal d'entrée est transféré à la sortie sans être modifié. Le but du filtre Transfert de composantes est de bricoler avec ce graphique simple à 45° pour vous permettre de changer la manière de relier les valeurs d'entrée aux valeurs de sortie.

Les mathématiques de base nous disent qu'une ligne droite comme celle-ci peut être définie par sa pente et le point où elle coupe l'axe des y. Une façon de modifier la conversion

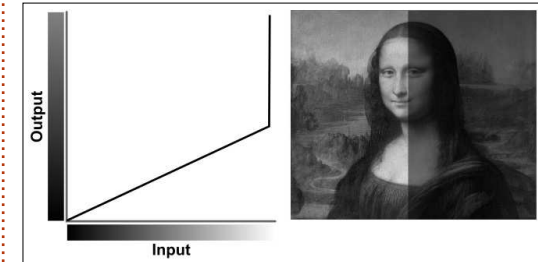
est donc de modifier la pente et le point d'intersection, un couple de valeurs fourni dans le filtre par l'option « Linéaire ». La ligne d'identité a une pente de 1, c'est-à-dire que, pour chaque augmentation de 1 sur l'axe des x, la valeur de y augmente de 1. En positionnant cette valeur à 2, nous accentuons la pente, entraînant une plus forte luminosité de la sortie. Voici à quoi ressemble le dialogue du filtre pour un canal :



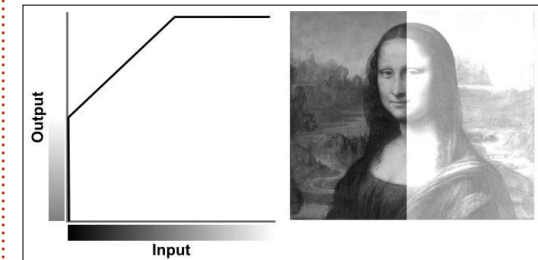
Tout en montrant l'effet sur la pente, j'ai aussi inclus une version de la Joconde en niveaux de gris, avec sur le côté droit le résultat de l'application de ce changement à tous les canaux de couleurs :



Le changement de la pente pour une valeur plus petite, 0,5 dans ce cas, réduit la luminosité de l'image :



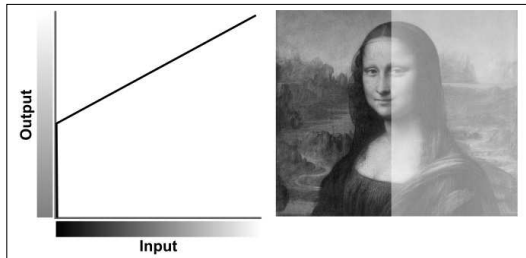
En changeant l'interception, vous pouvez modifier le contraste de l'image ; vous pourriez aussi vouloir ajuster la pente pour vous assurer que la luminosité n'est pas modifiée en même temps (sauf si c'est votre intention). Par exemple, en réglant l'interception à 0,5 avec une pente de 1, vous obtiendrez ce résultat :



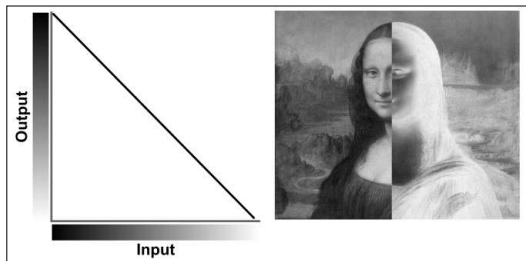
Gardez en tête que les canaux de couleurs ne peuvent pas passer sous 0 ou au-dessus de 127 ; aussi, le graphique change de forme quand vous atteignez ces limites. Comme vous



pouvez le voir, il devient horizontal à mi-chemin sur l'axe des x, délavant toutes les valeurs au-dessus de 127 en les rendant complètement blanches. Compenser ceci par une modification de la pente à 0,5 préserve beaucoup plus de détails, car les 255 valeurs sont toutes converties, au lieu d'en brider la moitié.

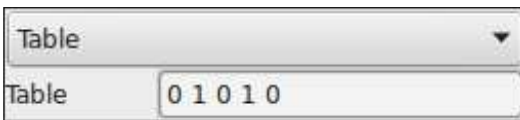


La valeur de l'interception peut aussi être négative, pour obtenir une sortie plus sombre, à nouveau avec un contraste réduit. Notez bien que la pente peut aussi être négative, ce qui inverse la conversion de sorte que les grandes valeurs sont changées en petites, et vice-versa. Avec une pente de -1 et une interception de 1, la sortie de ce canal est complètement inversée :

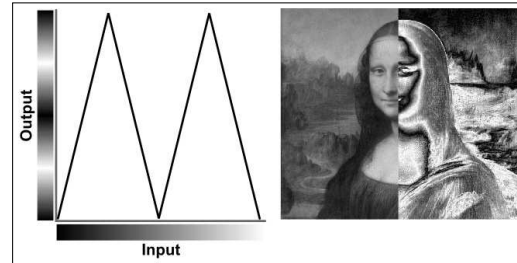


Le mode linéaire de la primitive de filtrage considère que vous voulez une simple conversion de l'entrée vers la sortie, pour ajuster la luminosité ou le contraste en modifiant la pente et la position d'une ligne unique. Mais, parfois, une ligne droite unique (même une qui est aplatie aux limites de la plage des couleurs) ne suffit pas. Qu'arrive-t-il si vous voulez que la sortie monte d'abord, puis redescende, de sorte que les valeurs d'extrémités sont converties en faibles valeurs, alors que celles du milieu sont remplacées par des valeurs élevées ? Pour cela, nous disposons du mode « Table ».

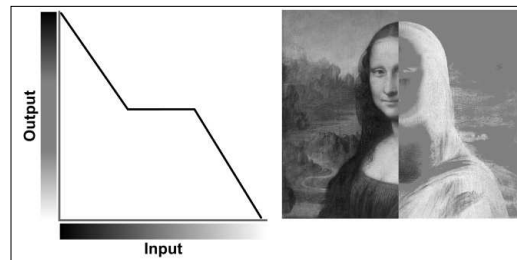
« Table » est un peu trompeur, car la table que vous devez fournir est mono-dimensionnelle. « Liste » aurait pu être un meilleur titre, mais table est ce que le groupe de travail SVG a décidé de choisir et ce qu'Inkscape applique. Les nombres de la liste représentent les valeurs de début et de fin d'une série de segments droits ; le nombre de valeurs de la liste détermine combien il y a de segments. Par exemple, la table ci-dessous a cinq valeurs (vous pouvez utiliser des espaces ou des virgules pour les séparer) :



Ces cinq valeurs donnent lieu à quatre segments distincts sur le graphique, entraînant une montée et une descente rapides des valeurs de sortie lorsque les valeurs d'entrée varient :



Une table constituée seulement de (0, 1) fera une conversion Identité, alors que (1, 0) inversera l'image. Pour aplatir une section de la ligne, l'utilisation de la même valeur deux fois de suite : (1, 0,5, 0,5, 0) donne une image inversée où les détails sont préservés dans les hautes et les basses valeurs, mais les nombres du tiers central sont tous convertis en 127 :



Comme vous pouvez le voir, l'étendue d'entrée est divisée également sur la base du nombre de valeurs de votre table et les lignes varient doucement entre elles. Quelquefois, cependant,

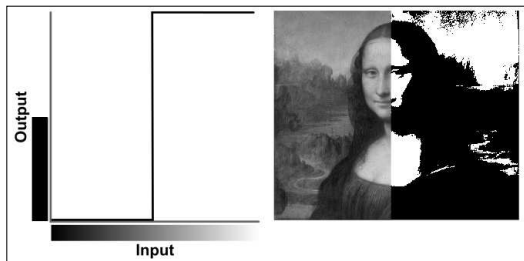
une transition douce n'est pas du tout ce que vous voulez. Supposez que vous devez réduire le nombre de couleurs de l'image (« postériser ») et même la ramener à une version en pur noir et blanc. Dans ces cas, utilisez le mode Discret.

En mode Discret, vous fournissez encore une « table » de valeurs, mais, plutôt que de définir des points de départ et d'arrivée entre lesquels une interpolation est calculée, vous fournissez une liste des seules valeurs de sortie autorisées et Inkscape les convertira en sections de la plage d'entrée. Fournissez seulement deux valeurs et toute valeur d'entrée de 127 ou moins sera convertie dans la première valeur, de 128 ou plus dans la deuxième. Monochrome instantané ! Fournissez 4 nombres et les valeurs de 0 à 63 seront transformées dans le premier nombre, de 64 à 127 dans le second, et ainsi de suite.

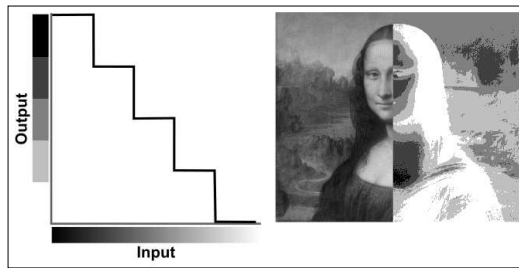
Sauf qu'il y a une erreur dans Inkscape qui l'empêche de fonctionner correctement. En mode Discret, la dernière valeur de votre liste est sautée. Ainsi, si vous fournissez deux valeurs en souhaitant obtenir une sortie monochrome, vous verrez que chaque valeur d'entrée est convertie dans la première valeur, et que la seconde n'est jamais utilisée. La solution est

évidemment de fournir trois nombres (typiquement en dupliquant le dernier), mais alors le filtre ne fonctionnera pas correctement dans d'autres programmes SVG ou navigateurs Web. Le défaut est suivi sur Launchpad sous l'erreur #1046093, et une solution a été validée pour la prochaine publication 0.92 d'Inkscape, ce qui est bien, mais cela veut dire aussi que, si vous fournissez une valeur supplémentaire pour faire fonctionner le filtre en 0.91, votre image sera faussée quand vous ferez la mise à niveau en 0.92.

Pour les exemples ci-dessous, j'ai fait comme si Inkscape fonctionnait correctement - gardez simplement en tête que ce que je dis (0, 1) devrait être réellement (0, 1, 1) pour que ça marche dans la version actuelle. A ce propos, voici cette version monochrome, en utilisant une table discrète contenant (0, 1) :



Celle-ci utilise les valeurs (1, 0.75, 0.5, 0.25, 0) pour postériser la Joconde en cinq nuances de gris, tout en inversant la sortie dans le même temps :



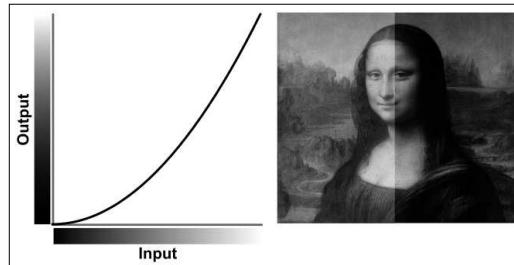
Une chose que vous aurez probablement déjà constaté pour tous ces modes, c'est que les graphiques sont entièrement constitués de lignes droites, horizontales dans le cas de Discret, anguleuses pour Table, Linéaire et Identité. La dernière option ajoute un peu de courbure au graphique, mais ne vous réjouissez pas trop ; elle ne vous permet pas de dessiner une courbe de Bézier quelconque, mais plutôt d'alimenter trois paramètres pour une courbe de correction du gamma.

Dans le cas où vous ne seriez pas familier avec la correction du gamma, c'est une conversion non linéaire des valeurs d'entrée vers celles de sortie, qui est utilisée pour ajuster la luminosité et le contraste d'une image, afin de compenser les différences de luminosité perçue aux deux bouts de la plage. Voyez-la comme une option plus sophistiquée que la simple modification de la pente et du point d'interception utilisés dans le mode Linéaire, car elle permet aux valeurs faibles d'être modifiées plus fortement que les fortes valeurs.

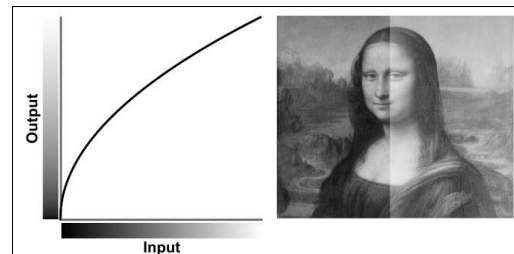
Le mode Gamma utilise trois paramètres : l'Amplitude, l'Exposant et l'Offset. La valeur de sortie de la fonction de transfert est calculée par la formule suivante :

$$\text{sortie} = \text{Amplitude} \times \text{entrée}^{\text{Exposant}} + \text{Offset}$$

C'est-à-dire que la valeur d'entrée est portée à la puissance de la valeur de l'exposant, multipliée par l'amplitude et ajoutée à l'offset. Souvent l'amplitude est maintenue à 1, et l'offset à 0, ce qui fait que la sortie est simplement l'entrée à la puissance de l'exposant. Ainsi, avec un exposant de 2, le résultat ressemble à ceci :



Pour éclaircir une image, utilisez simplement une valeur d'exposant inférieure à 1, tel cet exemple avec une valeur de 0,5.



Notez la similarité avec le mode Linéaire, avec les pentes à 0,5 (pour foncer) et 2 (pour éclaircir). Le mode Gamma offre souvent un résultat plus détaillé, particulièrement s'il y a des modifications subtiles dans les parties sombres de la plage d'entrée.

Bien que j'aie utilisé une image en niveaux de gris pour illustrer ce filtre, vous pouvez en pratique utiliser une fonction de transfert différente pour chaque composante de couleur, ainsi que pour le canal alpha, utile pour conserver intact le canal alpha dans le mode Identité alors que vous modifiez les canaux de couleurs, ou, au contraire, pour n'affecter que le canal alpha tout en conservant les couleurs intactes.

Donc, pour finir, voici une image de la Joconde dans toute sa gloire colorée, avec l'application de quatre différents transferts de composantes (page suivante). Le quart en haut à gauche est une Table (1, 0) appliquée au seul canal vert, avec les trois autres en Identité ; celui du haut à droite utilise une Table (0, 1, 0, 1, 0) sur tous les canaux de couleur ; en bas à droite, Discret (0, 0.25, 0.5, 0.75, 1) est utilisé sur les canaux de couleurs pour postériser l'image, et en bas à gauche, Table (1, 0) est utilisé sur tous les canaux pour produire un effet de « négatif photographique ».

Crédits image :  
*La Joconde* (alias *Mona Lisa*)  
par Leonardo da Vinci



[https://fr.wikipedia.org/wiki/Fichier:Mona Lisa, by Leonardo da Vinci, from C2RMF retouched.jpg](https://fr.wikipedia.org/wiki/Fichier:Mona_Lisa,_by_Leonardo_da_Vinci,_from_C2RMF_retouched.jpg)

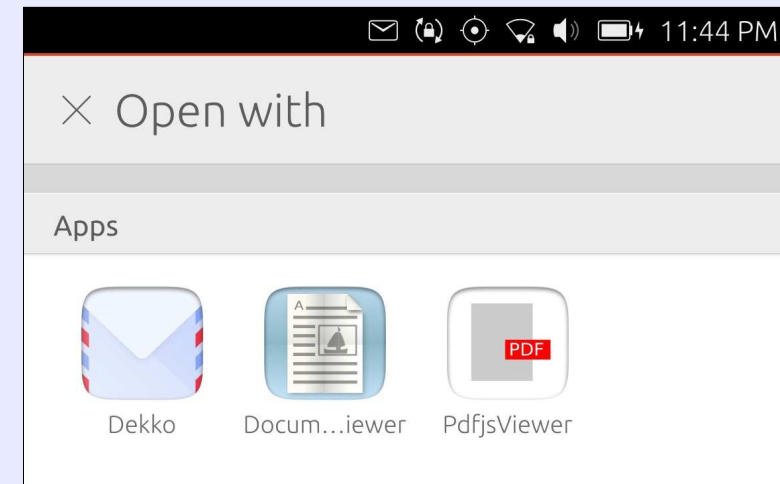
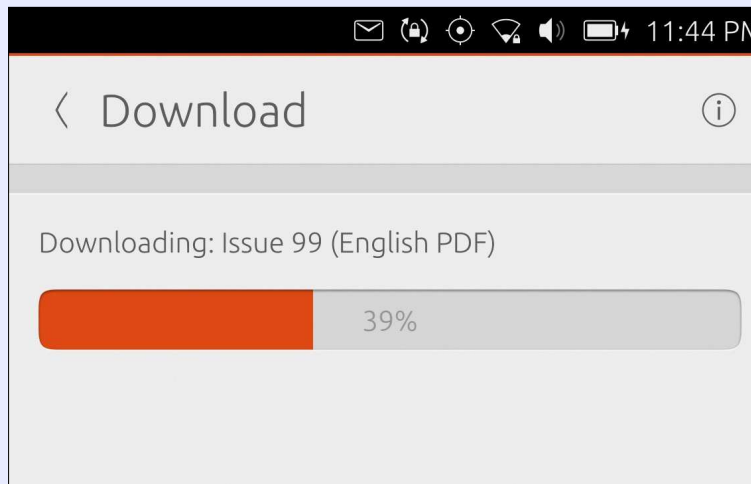
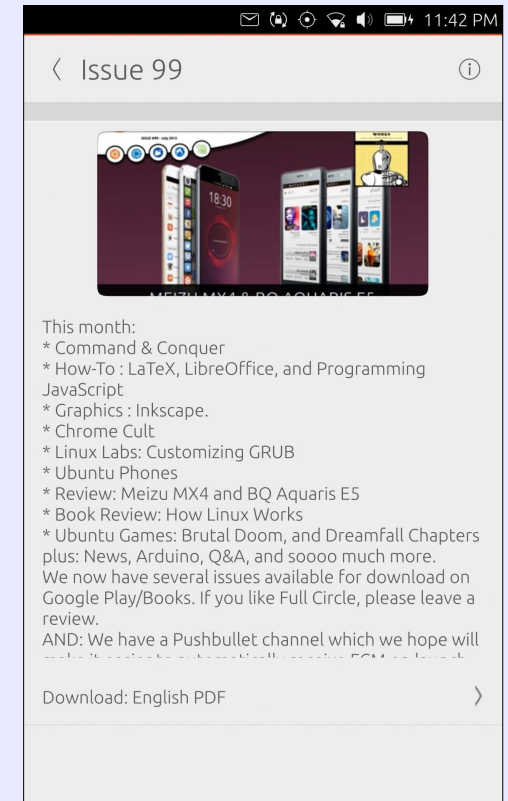
## L'APPLICATION OFFICIELLE FULL CIRCLE POUR UBUNTU TOUCH

Brian Douglass a créé une appli fantastique pour les appareils Ubuntu Touch, qui vous permettra de voir les numéros actuels et les numéros plus anciens, et de les télécharger et les lire sur votre téléphone/tablette Ubuntu Touch.

### INSTALLATION

Soit vous cherchez « full circle » dans le magasin Ubuntu Touch et vous cliquez sur Installer, soit vous affichez l'URL ci-dessous sur votre appareil et vous cliquez sur Installer pour être transféré sur la page des téléchargements.

<https://uappexplorer.com/app/fullcircle.bhdouglass>

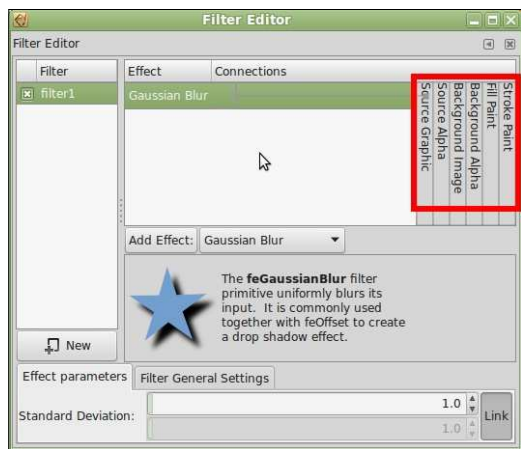


Mark a utilisé Inkscape pour créer trois bandes dessinées, *The Greys*, *Monsters*, *Inked* et *Elvie*, qui peuvent toutes être trouvées à <http://www.peppertop.com/>





Les filtres sont un sujet important pour tirer le meilleur parti d'Inkscape - au moins, pour des activités artistiques. Il y a toujours un danger avec le dessin vectoriel : qu'il puisse paraître, au final, trop précis et stérile pour certains usages ; les filtres sont souvent une façon de rajouter de fines (et pas si fines) variations dans la texture et la couleur qui sont souvent la marque de fabrique des dessins bitmap. Du moins, c'est la raison pour laquelle j'ai choisi de passer les neuf derniers articles de la série à vous présenter les filtres ; mais, ayant décrit chaque primitive dans le détail et montré quelques chaînes de filtrage ce faisant, cet article est le dernier sur ce sujet, et je passerai à autre chose le mois prochain.



Dans la partie 48, j'ai mentionné rapidement les colonnes d'entrées source sur la droite du dialogue des filtres (encadrées en rouge). Nous avons passé un peu de temps avec l'« Opacité de la source » et beaucoup plus avec la « Source image » ; mais, il reste quatre autres options qui ont été complètement ignorées jusqu'à maintenant. Il y a une bonne raison à cela, car dans la partie 48, j'ai écrit : « ...des six entrées visibles dans l'interface utilisateur, deux d'entre elles réclament un traitement spécial... et deux autres ne fonctionnent pas du tout ! »

Tout d'abord, réglons le sort des deux qui ne fonctionnent pas. « Remplissage » et « Remplissage du contour », d'après la spécification SVG, devraient faire exactement ce que suggèrent les noms. Elles devraient agir comme pour la primitive Remplissage, en remplissant la zone de filtrage avec une couleur, mais, plutôt que de spécifier la couleur dans la primitive de filtrage elle-même, elle est prise à partir de la couleur de remplissage ou de contour de l'objet sélectionné. Ceci semble être une façon brillante d'introduire une paire de couleurs dans votre chaîne de filtrage,

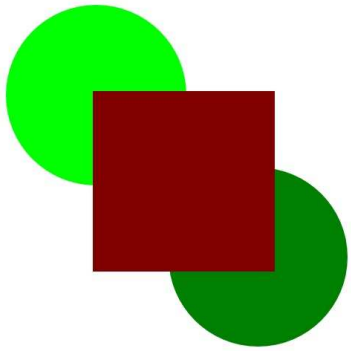
vous permettant de créer des filtres qui s'adaptent aux couleurs des objets auxquelles elles sont appliquées. Sauf que ça ne fonctionne pas du tout dans Inkscape.

Il y a un problème technique évident avec ces deux sources d'entrée : un remplissage ou un contour peut être plus qu'une simple couleur unie dans SVG. Ceci n'affecte pas vraiment leur utilisation dans une chaîne de filtres - un motif peut être répété pour remplir la zone de filtrage, de même qu'un dégradé, si la définition le permet - mais cela complique de manière significative le processus de rendu pour Inkscape, et cela n'a pas (encore) été abordé par les développeurs. Néanmoins, même la simple possibilité d'utiliser une couleur unie de remplissage ou de contour serait un ajout utile. S'ils ne projettent pas d'ajouter au moins cela, il y a longtemps que ces deux colonnes auraient dû être supprimées de l'interface pour éviter toute confusion.

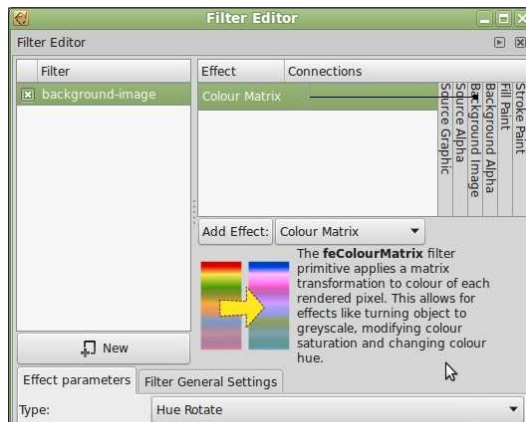
Les sources d'entrée restantes, « Image de fond » et « Opacité de fond », peuvent être utilisées dans Inkscape, mais seulement après une petite pré-

paration. Ces images représentent un « image instantanée du canevas sous la zone du filtrage au moment où l'élément de filtrage est invoqué » (d'après la spécification SVG). En d'autres termes, ils introduisent une image bitmap du dessin derrière la zone de filtrage (ou simplement le canal alpha de cette même zone) et la rendent disponible dans la chaîne de filtrage, comme l'introduction d'un bitmap via la primitive Image (ou presque). Cependant, la spécification indique aussi que la conservation d'une image de l'arrière-plan en mémoire « peut prendre une quantité significative des ressources système » ; aussi, le contenu SVG doit « indiquer explicitement » à l'application que le document doit avoir accès à l'arrière-plan avant que l'une de ces deux sources puisse avoir le moindre effet. Puis, elle définit comment spécifier à un document qu'il doit avoir accès à l'arrière-plan, en plaçant un attribut « enable-background » (activer l'arrière-plan) dans l'élément conteneur ascendant, lui donnant la valeur « new » (nouveau). Vous pouvez bricoler l'éditeur XML ou même modifier le code source du fichier dans un éditeur de texte, mais il y a une façon beaucoup plus facile de le faire.

Avant d'expliquer cette méthode plus simple, j'utiliserai un fichier texte très basique pour clarifier exactement ce dont je parle. Voici un couple de cercles verts qui seront mes objets d'arrière-plan. L'arrière-plan est constitué de tous les contenus qui sont sous l'objet filtré dans l'axe z ; ce peut être aussi bien une forme simple qu'un dessin complet. Devant les cercles se trouve un carré rouge, l'objet sur lequel j'appliquerai le filtre.

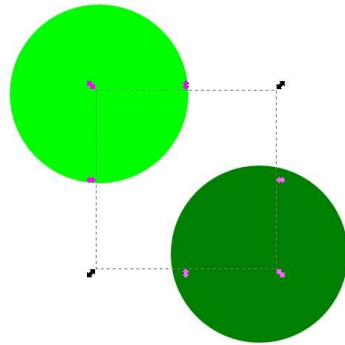


Le filtre lui-même est assez simple, juste une primitive Matrice de couleurs réglée sur le mode Décalage de teinte,

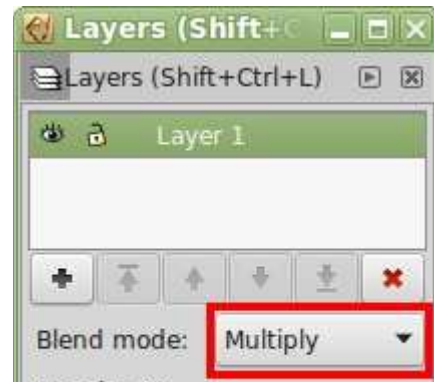


en utilisant l'Image de fond comme source.

Le résultat, à cet instant, est plutôt décevant. Le carré devient simplement transparent, sans aucun effet sur les cercles d'arrière-plan.

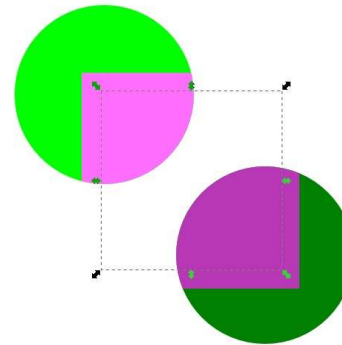


Maintenant, ajoutons l'attribut « enable-background ». Ouvrez simplement le dialogue des calques et changez le mode de fondu de l'un des calques pour autre chose que « Normal » (voyez la partie 9 de la série, si nécessaire). Ne paniquez pas si un effet inattendu apparaît sur votre image, car nous allons immédiatement revenir à « Normal ».



La magie devrait déjà avoir agi.

Mon image de test ressemble maintenant à ceci, avec les couleurs d'arrière-plan qui ont changé dans le secteur couvert par la zone de filtrage du carré. Par défaut, la région de filtrage s'étend bien au-delà de l'objet sélectionné ; c'est pourquoi le changement de couleur est présent à l'extérieur de la ligne pointillée de la boîte sélectionnée. Le carré lui-même a disparu, parce qu'il n'y a rien dans la chaîne de filtrage qui vient de la Source image.



Alors, que s'est-il passé ? Quelle magie noire, quel vaudou a fait que le filtre fonctionne juste en changeant un mode de fondu, puis en revenant immédiatement à la valeur initiale ? C'est vraiment très simple : le choix du mode de fondu dans le dialogue des calques est un raccourci pour ajouter un filtre avec la primitive Mélange à l'objet. Vous pouvez même voir qu'il apparaît dans l'éditeur de filtre.



Bien qu'il apparaisse dans le dialogue, le filtre n'est attaché à aucun objet que vous puissiez sélectionner sur le canevas. Il est plutôt lié au calque lui-même. Souvenez-vous que les calques sont juste un groupe auquel sont ajoutés quelques attributs spécifiques à Inkscape ; aussi, ce n'est pas vraiment différent d'un filtre appliqué à un groupe d'objets. Quand le filtre est créé, Inkscape connecte automatiquement les entrées de la primitive Mélange à la Source image (c'est-à-dire le calque qui est vraiment un groupe) et à l'Image de fond. En même temps, il ajoute l'attribut « enable-background » au nœud racine du document SVG, visible dans l'éditeur XML d'Inkscape (page suivante).

Le point clé est que le retour du Mode de rendu à « Normal » laisse cet attribut intact, alors qu'il a bien retiré le filtre. A partir de ce moment-là, vous êtes libre d'utiliser les entrées

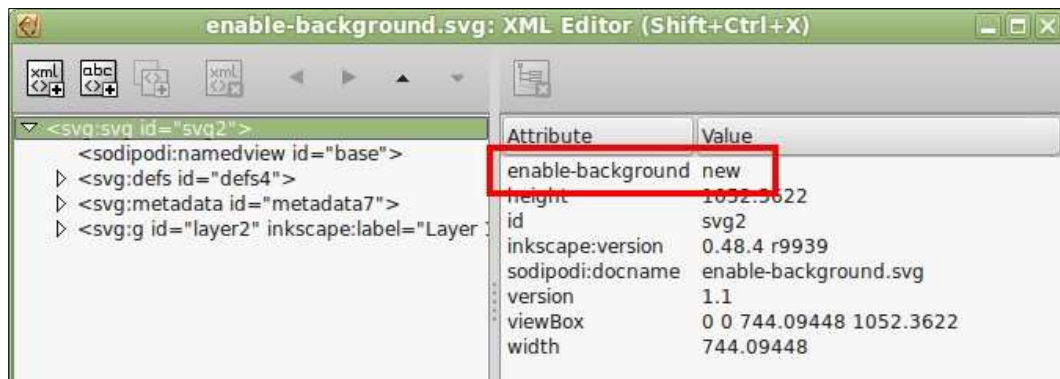


Image de fond et Opacité de fond dans vos propres chaînes de filtrage.

Ceci conclut notre examen détaillé de l'art mystérieux de la création de vos propres chaînes de filtrage. Mais, si vous avez fait des tests, vous avez noté, sans aucun doute, qu'Inkscape fournit déjà une liste étendue de filtres prêts à l'emploi, groupés par type, qui représentent la majeure partie du menu des Filtres. Malgré l'existence de ces courageux masochistes qui osent braver les défaillances de l'interface utilisateur d'Inkscape pour créer leurs propres filtres complexes « ex nihilo », beaucoup plus d'utilisateurs travaillent simplement avec l'ensemble fourni par défaut. Mais, avec le savoir que vous avez engrangé ces derniers mois, vous pouvez faire mieux que ça : vous pouvez commencer avec un filtre standard, puis plonger dans ses entrailles pour l'éditer et l'ajuster pour répondre à vos besoins.

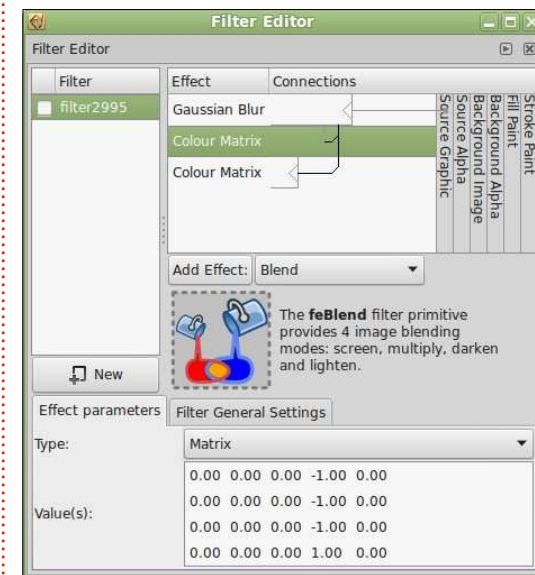
Je ne perdrai pas de temps à passer exhaustivement en revue la liste des filtres par défaut, mais, à la place, je vous encourage à les explorer vous-même. Essayez de créer une feuille de test avec différents objets et groupes pour travailler dessus : certains filtres fonctionnent mieux avec des petits objets, d'autres avec des grands ; certains ont besoin d'un contenu coloré, d'autres marchent avec une simple forme monochrome ; certains sont perdus avec des contours compliqués, pendant que d'autres sont en défaut sur de simples blocs de couleur sans fioritures.

Une possibilité classique d'Inkscape est que, quand vous copiez/collez un objet d'un document à un autre, tous les filtres attachés le suivent. Pourquoi ne pas commencer une « bibliothèque de filtres », un document dans lequel vous copiez tout filtre particulièrement utile ou impressionnant ? Chaque fois que vous créez ou trou-

vez un filtre génial, appliquez-le simplement à un objet approprié, puis copiez/collez-le dans le fichier de la bibliothèque. De même, quand vous voulez utiliser un filtre de votre bibliothèque, copiez juste l'objet du fichier de la bibliothèque et collez-le dans votre dessin en cours de création. Le filtre apparaîtra dans le dialogue des filtres et, une fois appliqué à quelque chose d'autre dans votre image, vous pourrez effacer sans risque l'objet tiré de la bibliothèque. Des utilisateurs ont déjà posté en ligne leurs propres collections de filtres ; cherchez « Inkscape filter pack » (paquet de filtres Inkscape), par exemple. Vous pourriez trouver ainsi que quelqu'un d'autre a déjà créé le filtre dont vous avez besoin, et il vous suffit d'un copier/coller pour l'utiliser dans vos propres dessins.

Quand vous créez vos propres chaînes de filtrage à partir de rien, la question sur ce qui se passe en combinant deux primitives ne se pose pratiquement jamais. Vous voulez un filtre pour flouter et désaturer ? Pas de problème, simplement enchaînez une primitive Flou gaussien avec une primitive Matrice de couleurs (en mode Saturation). Mais qu'arrive-t-il si vous voulez faire la même chose avec les filtres par défaut ? Il y a, dans le menu Filtres, Flou > Flou ( qui fournit une simple primitive Flou gaussien) et aussi

Couleurs > Brillance (qui fournit une simple primitive Matrice de couleurs). Que se passe-t-il si vous ajoutez les deux à un objet ? Si vous l'essayez, vous verrez que vous obtenez un résultat flou et désaturé ; ainsi, il est possible de combiner de cette manière les filtres par défaut. Mais il se passe un truc bizarre dans la chaîne de filtres. Vous n'avez pas seulement les deux primitives que vous avez prévu, mais aussi une troisième : une Matrice de couleurs supplémentaire entre les deux primitives attendues.



Si vous regardez avec attention, vous verrez qu'elle n'est même pas reliée à la dernière primitive ; elle ne joue donc aucun rôle actif dans la chaîne. Vous pouvez la supprimer et cela n'aura aucun effet. Alors, pourquoi est-elle là ?



Il s'avère que c'est un ajout plutôt sympa de la part des développeurs d'Inkscape. Supposons que vous vouliez ajouter un autre filtre à cette chaîne, mais que ça en soit un qui utiliserait normalement l'Opacité de la source comme entrée. Pour empêcher tout résultat inattendu, vous avez besoin qu'il utilise l'alpha à partir de la sortie du filtre précédent, qui ne sera en général pas du tout pareil que l'Opacité de la source elle-même. Ces primitives Matrice de couleurs supplémentaires agissent comme des sorties alpha intermédiaires dans la chaîne. Ainsi, avec l'ajout de celles-ci, vous pouvez non seulement lier toute nouvelle primitive à l'image de sortie de chaque filtre, mais aussi à la sortie alpha.

Pouvoir combiner des filtres de cette manière et les voir fonctionner comme prévu est impressionnant, mais cela peut conduire rapidement à de longues chaînes de filtrage compliquées et difficiles à gérer. Une meilleure approche est souvent d'appliquer un filtre, puis de grouper votre objet avant d'appliquer le filtre suivant au groupe. Vous pouvez répéter ceci autant de fois que nécessaire, créant un emboîtement toujours plus long de groupes, chacun ayant son propre filtre. Ceci rend certainement plus facile leur gestion dans le dialogue des filtres, car il y a beaucoup moins de con-

fusion sur quel filtre vous êtes en train de modifier, particulièrement si vous les nommez bien.

Une dernière chose à noter est que, dans la 0.91 (et dans la 0.92 qui vient de sortir !), beaucoup de filtres par défaut ont des points de suspension (trois points « ... ») en fin d'appellation. Le choix de l'un d'eux ouvre une boîte de dialogue qui vous permet de saisir les paramètres du filtre et voir une prévisualisation immédiate. Bien sûr, c'est juste un raccourci pour régler les paramètres de chaque primitive de filtrage, mais c'est une addition bienvenue qui peut présenter les paramètres les plus importants de nombre de primitives, tout en cachant les autres options et réglages qui n'ont aucun intérêt dans la plupart des cas. Malheureusement, vous n'avez aucune possibilité de retrouver cette interface simplifiée une fois que vous l'avez renvoyée ; aussi, tout réglage ultérieur nécessitera de plonger à nouveau dans tous les détails sordides de chaque primitive. Certains filtres distincts venant de la 0.48 ont été supprimés, puisque les nouveaux filtres paramétrés peuvent créer les mêmes effets, voire plus. Si vous ne trouvez pas votre vieux filtre favori dans les nouvelles versions, cherchez un nom voisin avec trois points de suspension et commencez à ajuster les paramètres !



Mark a utilisé Inkscape pour créer trois bandes dessinées, *The Greys*, *Monsters*, *Inked* et *Elvie*, qui peuvent toutes être trouvées à <http://www.peppertop.com/>

## ENQUÊTE 2018 DU FULL CIRCLE

C'est à nouveau la période de l'année où nous vous demandons ce que vous pensez du FCM, d'Ubuntu et de Linux.

Certaines questions sont obligatoires, d'autres peuvent être omises si elles ne vous concernent pas.

Vos réponses aideront à façonner le Full Circle pour l'année à venir, alors, s'il vous plaît, utilisez votre sens critique de façon constructive. Si vous ne nous dites pas ce que vous pensez, ou ce que nous faisons mal, nous ne le saurons pas.

URL de l'enquête :

<http://bit.ly/fcm2018>





## Lignes directrices

**N**otre seule règle : tout article doit avoir un quelconque rapport avec Ubuntu ou avec l'une de ses dérivées (Kubuntu, Xubuntu, Lubuntu, etc.).

## Autres règles

- Les articles ne sont pas limités en mots, mais il faut savoir que de longs articles peuvent paraître comme série dans plusieurs numéros.

- Pour des conseils, veuillez vous référer au guide officiel *Official Full Circle Style Guide* ici : <http://url.fullcirclemagazine.org/75d471>

- Utilisez n'importe quel logiciel de traitement de texte pour écrire votre article – je recommande LibreOffice –, mais le plus important est d'en **VÉRIFIER L'ORTHOGRAPHE ET LA GRAMMAIRE !**

- Dans l'article veuillez nous faire savoir l'emplacement souhaité pour une image spécifique en indiquant le nom de l'image dans un nouveau paragraphe ou en l'intégrant dans le document ODT (OpenOffice/LibreOffice).

- Les images doivent être en format JPG, de 800 pixels de large au maximum et d'un niveau de compression réduit.

- Ne pas utiliser des tableaux ou toute sorte de formatage en **gras** ou *italique*.

Lorsque vous êtes prêt à présenter l'article, envoyez-le par courriel à : [articles@fullcirclemagazine.org](mailto:articles@fullcirclemagazine.org).

*Si vous écrivez une critique, veuillez suivre ces lignes directrices :*

## Traductions

Si vous aimeriez traduire le Full Circle dans votre langue maternelle, veuillez envoyer un courriel à [ronnie@fullcirclemagazine.org](mailto:ronnie@fullcirclemagazine.org) et soit nous vous mettrons en contact avec une équipe existante, soit nous pourrons vous donner accès au texte brut que vous pourrez traduire. Lorsque vous aurez terminé un PDF, vous pourrez télécharger votre fichier vers le site principal du Full Circle.

## Auteurs francophones

Si votre langue maternelle n'est pas l'anglais, mais le français, ne vous inquiétez pas. Bien que les articles soient encore trop longs et difficiles pour nous, l'équipe de traduction du FCM-fr vous propose de traduire vos « Questions » ou « Courriers » de la langue de Molière à celle de Shakespeare et de vous les renvoyer. Libre à vous de la/les faire parvenir à l'adresse mail *ad hoc* du Full Circle en « v.o. ». Si l'idée de participer à cette nouvelle expérience vous tente, envoyez votre question ou votre courriel à :

[webmaster@fullcirclemag.fr](mailto:webmaster@fullcirclemag.fr)

## Écrire pour le FCM français

Si vous souhaitez contribuer au FCM, mais que vous ne pouvez pas écrire en anglais, faites-nous parvenir vos articles, ils seront publiés en français dans l'édition française du FCM.

## CRITIQUES

### Jeux/Applications

**Si vous faites une critique de jeux ou d'applications, veuillez noter de façon claire :**

- le titre du jeu ;
- qui l'a créé ;
- s'il est en téléchargement gratuit ou payant ;
- où l'obtenir (donner l'URL du téléchargement ou du site) ;
- s'il est natif sous Linux ou s'il utilise Wine ;
- une note sur cinq ;
- un résumé avec les bons et les mauvais points.

### Matériel

**Si vous faites une critique du matériel veuillez noter de façon claire :**

- constructeur et modèle ;
- dans quelle catégorie vous le mettriez ;
- les quelques problèmes techniques éventuels que vous auriez rencontrés à l'utilisation ;
- s'il est facile de le faire fonctionner sous Linux ;
- si des pilotes Windows ont été nécessaires ;
- une note sur cinq ;
- un résumé avec les bons et les mauvais points.

**Pas besoin d'être un expert pour écrire un article ; écrivez au sujet des jeux, des applications et du matériel que vous utilisez tous les jours.**





# MÉCÈNES

## DONS MENSUELS

### 2016 - Present:

Bill Berninghausen  
 Jack McMahon  
 Linda P  
 Remke Schuurmans  
 Norman Phillips  
 Tom Rausner  
 Charles Battersby  
 Tom Bell  
 Oscar Rivera  
 Alex Crabtree  
 Ray Spain  
 Richard Underwood  
 Charles Anderson  
 Ricardo Coalla  
 Chris Giltane  
 William von Hagen  
 Mark Shuttleworth  
 Juan Ortiz  
 Joe Gulizia  
 Kevin Raulins  
 Doug Bruce  
 Pekka Niemi  
 Rob Fitzgerald  
 Brian M Murray  
 Roy Milner  
 Brian Bogdan  
 Scott Mack  
 Dennis Mack  
 John Helmers

JT  
 Elizabeth K. Joseph  
 Vincent Jobard  
 Chris Giltane  
 Joao Cantinho Lopes  
 John Andrews

### 2017 - Present:

Matt Hopper  
 Jay Pee  
 Brian Kelly  
 J.J. van Kampen

### 2018 - Present:

John Helmers  
 Kevin O'Brien  
 Kevin Raulins  
 Carl Andersen  
 Charles Stewart  
 Dave Nelson  
 Brian Bogdan

## DONS UNIQUES

### 2017:

Linda Prinsen  
 Shashank Sharma  
 Glenn Heaton  
 Frank Dinger  
 Randy E. Brinson  
 Kevin Dwyer  
 Douglas Brown

Daniel Truchon  
 John Helmers  
 Ronald Eike  
 Dennis Shimer  
 Iain Mckeand  
 Jaideep Tibrewala  
 Kevin Dwyer

### 2018:

Yvo Geens  
 Graig Pearen  
 Carlo Puglisi  
 James A Carnrite  
 John Holman  
 P G Schmitt  
 Robert Cannon  
 Thomas A Lawell  
 Ronald Le Blanc  
 Luis Eduardo Herman  
 Glenn Heaton  
 Peter Swentzel  
 Alain Mallette  
 Christophe Caron  
 Linda Prinsen  
 Ronald Eike  
 Anthony Cooper  
 Louis W Adams Jr  
 Joseph Tong

Le site actuel du Full Circle Magazine fut créé grâce à **Lucas Westermann** (Monsieur Command & Conquer) qui s'est attaqué à la reconstruction entière du site et des scripts à partir de zéro, pendant ses loisirs.

La page Patreon (Mécènes) existe pour aider à payer les frais du domaine et de l'hébergement. L'objectif annuel fut rapidement atteint grâce à ceux dont le nom figure sur cette page. L'argent contribue aussi à la nouvelle liste de diffusion que j'ai créé.

Parce que plusieurs personnes ont demandé une option PayPal (pour un don ponctuel), j'ai ajouté un bouton sur le côté droit du site Web.

De très sincères remerciements à tous ceux qui ont utilisé Patreon et le bouton PayPal. Leurs dons m'aident ÉNORMÉMENT.



<https://www.patreon.com/fullcirclemagazine>



<https://paypal.me/ronnietucker>



<https://donorbox.org/recurring-monthly-donation>







# COMMENT CONTRIBUER

## FULL CIRCLE A BESOIN DE VOUS !

Un magazine n'en est pas un sans articles et Full Circle n'échappe pas à cette règle. Nous avons besoin de vos opinions, de vos bureaux et de vos histoires. Nous avons aussi besoin de critiques (jeux, applications et matériels), de tutoriels (sur K/X/L/Ubuntu), de tout ce que vous pourriez vouloir communiquer aux autres utilisateurs de \*buntu. Envoyez vos articles à :

[articles@fullcirclemagazine.org](mailto:articles@fullcirclemagazine.org)

Nous sommes constamment à la recherche de nouveaux articles pour le Full Circle. Pour de l'aide et des conseils, veuillez consulter l'Official Full Circle Style Guide :

<http://url.fullcirclemagazine.org/75d471>

Envoyez vos **remarques** ou vos **expériences** sous Linux à : [letters@fullcirclemagazine.org](mailto:letters@fullcirclemagazine.org)

Les tests de **matériels/logiciels** doivent être envoyés à : [reviews@fullcirclemagazine.org](mailto:reviews@fullcirclemagazine.org)

Envoyez vos **questions** pour la rubrique Q&R à : [questions@fullcirclemagazine.org](mailto:questions@fullcirclemagazine.org)

et les **captures d'écran** pour « Mon bureau » à : [misc@fullcirclemagazine.org](mailto:misc@fullcirclemagazine.org)

Si vous avez des questions, visitez notre forum : [fullcirclemagazine.org](http://fullcirclemagazine.org)

**FCM n° 134**



**Date de parution du numéro en langue anglaise :**

Vendredi 29 juin 2018.

Équipe Full Circle



**Rédacteur en chef** - Ronnie Tucker  
[ronnie@fullcirclemagazine.org](mailto:ronnie@fullcirclemagazine.org)

**Webmaster** - Lucas Westermann  
[admin@fullcirclemagazine.org](mailto:admin@fullcirclemagazine.org)

**Correction et Relecture**

Mike Kennedy, Gord Campbell, Robert Orsino, Josh Hertel, Bert Jerred, Jim Dyer et Emily Gonyer

Remerciements à Canonical, aux nombreuses équipes de traduction dans le monde entier et à **Thorsten Wilms** pour le logo du FCM.

**Pour la traduction française :**

<http://www.fullcirclemag.fr>

**Pour nous envoyer vos articles en français pour l'édition française :**

[webmaster@fullcirclemag.fr](mailto:webmaster@fullcirclemag.fr)

## Obtenir le Full Circle Magazine :

### Pour les Actus hebdomadaires du Full Circle :



Vous pouvez vous tenir au courant des Actus hebdomadaires en utilisant le flux RSS : <http://fullcirclemagazine.org/feed/podcast>



Ou, si vous êtes souvent en déplacement, vous pouvez obtenir les Actus hebdomadaires sur Stitcher Radio (Android/iOS/web) :

<http://www.stitcher.com/s?fid=85347&refid=stpr>



et sur Tunein à : <http://tunein.com/radio/Full-Circle-Weekly-News-p855064/>



**Format EPUB** - Les éditions récentes du Full Circle comportent un lien vers le fichier epub sur la page de téléchargements. Si vous avez des problèmes, vous pouvez envoyer un courriel à : [mobile@fullcirclemagazine.org](mailto:mobile@fullcirclemagazine.org)



**Issuu** - Vous avez la possibilité de lire le Full Circle en ligne via Issuu : <http://issuu.com/fullcirclemagazine>. N'hésitez surtout pas à partager et à noter le FCM, pour aider à le faire connaître ainsi qu' Ubuntu Linux.



**Magzster** - Vous pouvez aussi lire le Full Circle online via Magzster : <http://www.magzster.com/publishers/Full-Circle>. N'hésitez surtout pas à partager et à noter le FCM, pour aider à le faire connaître ainsi qu'Ubuntu Linux.

### Obtenir le Full Circle en français :

<http://www.fullcirclemag.fr/?pages/Numéros>

