



1– Introduction

- 1– [Introduction](#)
- 2– [Mise au point capitale](#)
- 3– [Le pourquoi du comment](#)
- 4– [Les moyens](#)
 - 4.1 [Killustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5– [Réalisation de l'oeil](#)
- 6– [Conclusion](#)

Pendant longtemps, le dessin vectoriel a été synonyme d'illustrations toujours précises quant aux contours, mais désespérément "plates". La raison en incombait à l'impossibilité de remplir des formes tracées autrement que par des aplats de couleur ou, à la rigueur, par des motifs bitmap. Il en résultait un aspect BD, vraiment pas réaliste, ce qui, par ailleurs, peut être un effet recherché. Mais dès que l'on veut produire une représentation de la réalité la plus exacte possible, il faut tenir compte du fait que les objets ont une épaisseur et qu'ils sont rarement vus dans un plan parallèle à leur face la plus significative. Les couleurs perçues de l'objet ne sont donc pas uniformes; elles varient de clair à sombre dans la même teinte –la couleur proprement dite selon la définition du modèle TSL–, ou même d'une teinte à une autre, du rouge vers bleu par exemple. Et si l'objet croise une zone d'ombre, la partie colorée concernée verra sa luminosité diminuer c'est à dire, toujours selon le modèle TSL, que la couleur perçue devra contenir un pourcentage de noir plus ou moins important.

Les programmes de dessins vectoriels d'aujourd'hui peuvent rendre compte de ces variations et permettre ainsi une représentation honnête de la réalité. C'est le sens qui par la suite sera donné à l'expression **rendu réaliste**.

[Retour à la section 2D](#)



2- Mise au point capitale

- 1- [Introduction](#)
- 2- [Mise au point capitale](#)
- 3- [Le pourquoi du comment](#)
- 4- [Les moyens](#)
 - 4.1 [Killustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5- [Réalisation de l'oeil](#)
- 6- [Conclusion](#)

Toutefois, il serait erroné de penser à la lecture des lignes précédentes qu'une fonctionnalité miracle existerait, et qu'il suffirait de l'activer pour voir une simple forme de Bézier devenir une citrouille hyper réaliste, pas plus que ladite citrouille ne deviendra un carrosse pour Cendrillon sous l'effet d'un coup de baguette magique...

Il s'agira en fait de tromper le regard en simulant le réel avec les outils dont nous disposons. Nous percevons les volumes parce que leurs couleurs, censées être identiques en tout point, ne sont pas unies: ou elles changent brusquement, et nous savons alors qu'une face de l'objet est oblique par rapport à une autre; ou bien les couleurs changent graduellement, et nous savons alors que notre oeil parcourt une surface courbe.

Le programme de dessin capable de rendre cet aspect des choses disposera donc de la fonction: "**Dégradé de couleur**". Trente pour cent du secret du rendu réaliste réside dans cette aptitude à donner du modelé à une forme, le pourcentage restant étant à partager entre le choix de la couleur (la teinte) et la direction supposée de l'éclairage de la scène, qui est un élément trop souvent négligé.

Les peintres procèdent de la même façon. Le volume d'un objet est rendu par une juxtaposition d'aplats de couleurs issues de la même teinte de base. Selon le style et la technique propres à un artiste, les juxtapositions peuvent être franches ou alors très subtiles, tel le sfumato de Léonard de Vinci, qui désigne une transition douce d'une couleur vers une autre. Les pastellistes, eux, recourront plus particulièrement à l'estompage avec le même objectif en vue.

Les programmes de dessin actuels permettent les transitions douces des peintres: ils permettent donc le rendu réaliste d'une image.

Mais les peintres procèdent également par superposition de couleurs. En utilisant ainsi la transparence plus ou moins grande des matières colorantes, ils augmentent la richesse de leur palette. C'est particulièrement vrai dans la technique de l'aquarelle et, à un degré moindre, de la gouache.

Qu'en est-il des programmes de dessin numérique dans ce domaine?

Pour être clair, disons que tout logiciel qui veut être respecté, bitmap ou vectoriel, doit gérer la transparence: c'est une fonction très importante pour ce que nous nous proposons d'atteindre.

Alors on l'aura compris: le réalisme, c'est l'infographiste, comme le peintre, qui le confère à l'image réalisée, et non l'ordinateur et son programme. C'est l'infographiste avec deux outils et même trois:

- le **dégradé de couleurs**, extrêmement courant.
- la **transparence** bien gérée, beaucoup moins courant.
- l'**estompage**, rarissime.

[Retour à la section 2D](#)



3- Le pourquoi du comment

- 1- [Introduction](#)
- 2- [Mise au point capitale](#)
- 3- [Le pourquoi du comment](#)
- 4- [Les moyens](#)
 - 4.1 [Killustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5- [Réalisation de l'oeil](#)
- 6- [Conclusion](#)

Depuis janvier 2000, j'effectue des bêta-tests sur **Artstream**, de Mediascape; mais depuis plus longtemps encore, je suis utilisateur enthousiaste de **Sketch**, de chez GPL;-), écrit par Bernhard Herzog. Dans ce cadre de test, il m'a été donné de réaliser une illustration similaire avec les deux programmes.



[Retour à la section 2D](#)

Les résultats sont proches et cependant une image est plus réaliste que l'autre: c'est particulièrement visible sur les illustrations à l'échelle 1 ou supérieure.

Pourquoi donc?

Dans un portrait, le réalisme, donc la ressemblance, est donné par le regard; le restant des traits pourraient être esquissé, si le regard est saisi, la ressemblance est assurée. Supposons qu'il en soit de même pour les Tuxy (sans existence réelle, ce n'est qu'une hypothèse d'école), et que le réalisme doive porter sur les yeux.

Dans un premier temps, il semble intéressant de faire l'inventaire des possibilités de dessin vectoriel sous Linux. Dressé en mai 2000, ce tour d'horizon ne saurait être exhaustif, puisque Corel Draw9, entre autre, qui s'annonce n'a pas été testé ici...



4- Les moyens

- 1- [Introduction](#)
- 2- [Mise au point capitale](#)
- 3- [Le pourquoi du comment](#)
- 4- [Les moyens](#)
 - 4.1 [Killustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5- [Réalisation de l'oeil](#)
- 6- [Conclusion](#)

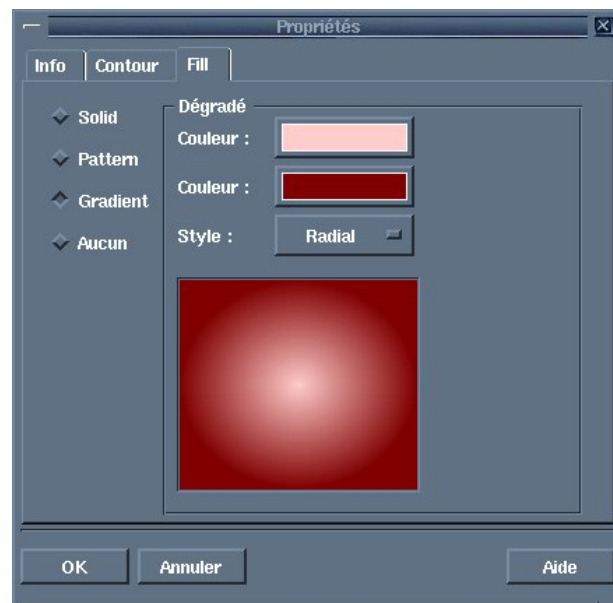
Pour comparer les logiciels, il a été tracé sur chacun d'eux cercle rempli d'un dégradé de couleur en vue de représenter une sphère le plus réaliste possible. Le résultat obtenu est montré ainsi que l'oeil du Tuxy qu'il est possible d'obtenir

4.1- Killustrator

Inclus dans les paquetages KDE, chacun dispose de cet outil en version 0.7x. Rapide, parfaitement intégré dans son environnement, il s'avère à l'usage stable, mais très limité. On est encore loin de l' *Illustrator Killer* annoncé lors de sa sortie. Les réglages se limitent au style (Horizontal, Vertical, Radial, Circulaire) et au choix des deux couleurs. De plus, la transparence n'existe pas. Le résultat est une sphère honnêtement sphérique, recevant un éclairage impossible à modifier et disposé à la place de l'observateur.

Pour l'heure, ce programme en cours de développement n'offre pas suffisamment de fonctionnalités pour être utilisé dans une optique de rendu réaliste. Pas d'oeil de Tuxy.

[Retour à la section 2D](#)





- 1- [Introduction](#)
- 2- [Mise au point capitale](#)
- 3- [Le pourquoi du comment](#)
- 4- [Les moyens](#)
 - 4.1 [Killustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5- [Réalisation de l'oeil](#)
- 6- [Conclusion](#)

De plus, Star Draw dispose d'une option transparence. Mais il faut bien admettre que celle-ci est gérée grossièrement. En fait de transparence, Star Draw ôte 0%, 25%, 50%, 75% 100% des pixels remplissant une forme; ce qui fait que l'on voit l'objet situé en dessous non en transparence mais comme à travers une passoire ayant plus ou moins de trous. Cependant, dans le dessin des deux planètes réalisé en trois clics de souris, l'ombre portée qui a subi ce traitement de transparence se montre assez réaliste.



[Retour à la section 2D](#)



4.2- Star Draw

- 1- [Introduction](#)
- 2- [Mise au point capitale](#)
- 3- [Le pourquoi du comment](#)
- 4- [Les moyens](#)
 - 4.1 [Killustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5- [Réalisation de l'oeil](#)
- 6- [Conclusion](#)

Voici l'étoile oubliée, injustement méconnue. Partie intégrante de Star Office, ce composant disparaît derrière la notoriété du traitement de texte ou du tableur de la suite. Cependant, Star Draw n'a pas son pareil pour réaliser proprement des organigrammes, des schémas très précis, gérer des blocs de textes, produire des dessin en 3D avec rendu OpenGL....

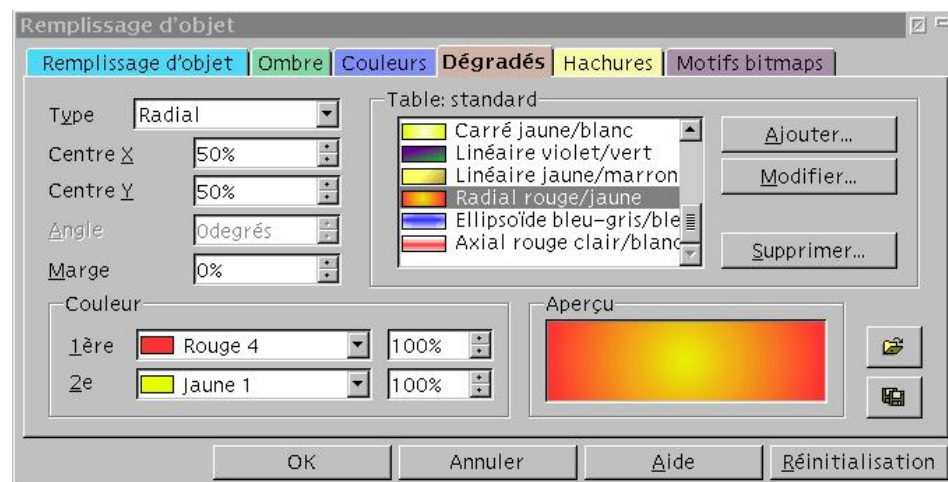
A découvrir d'urgence. Celui qui cherche un outil facile à mettre en oeuvre pour réaliser des pages mêlant texte, dessin, graphique, comme des plaquettes, des affiches, des cours... dispose là de l'outil adéquat.

Il eût été facile de réaliser une sphère réaliste avec Star Draw: la sphère est une primitive 3D sur laquelle le logiciel peut plaquer des textures, et diriger des éclairages multiples, colorés ou non... Les dès eussent été pipés. Aussi, une sphère a-t'elle été simulée à partir d'un cercle, comme sur les autres programmes de l'essai.

La gestion du dégradé de couleur est presque complète dans une boîte de dialogue à laquelle on accède par un clic droit sur le cercle que l'on vient de tracer. Le choix du type (Radial, linéaire etc..), de couleurs prédéfinies dans une table standard ou personnelle, de la position de l'éclaircement par % de X et Y, de la dimension du spéculaire exprimée en % de marge, des couleurs différentes des couleurs prédéfinies... est à disposition de l'utilisateur. Comme ces réglages peuvent s'avérer fastidieux, le système propose systématiquement une sauvegarde de ceux-ci.

Le résultat est intéressant, rapide et reproductible (ce qui est rare)

[Retour à la section 2D](#)



[Suite](#)

[==>](#)



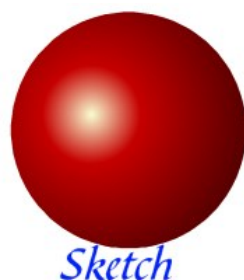
4.3- Sketch

- 1- [Introduction](#)
- 2- [Mise au point capitale](#)
- 3- [Le pourquoi du comment](#)
- 4- [Les moyens](#)
 - 4.1 [Killustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5- [Réalisation de l'oeil](#)
- 6- [Conclusion](#)

Ecrit en Python, Sketch pourrait faire figure de petit programme bidouillé en langage interprété, donc lent et peu puissant. Erreur: c'est tout le contraire. Même en version 0.6.6, Sketch est rapide, performant et remarquablement stable. Pas une fois et après beaucoup d'illustrations réalisées je n'ai eu de crash à déplorer. La gestion de dégradés est presque parfaite. Par rapport aux deux programmes cités plus avant, les dégradés ne sont pas limités en nombre de couleurs. Un clic droit sur la barre "Edition de dégradé" permet l'ajout d'un point de couleur; un clic droit sur le point inséré permet de lui affecter une couleur, ou de modifier une existante. Ce point se déplace à la souris sur la barre pour affecter un % plus important de couleur concernée.

Quant à la position du spéculaire, on le règle à la souris dans la fenêtre de prévisualisation de la boîte "Style de remplissage". On sent bien que le grand Illustrator a servi de modèle; c'eût été encore mieux si le modèle avait été Canvas.

[Retour à la section 2D](#)



Mais le résultat est très réaliste; les dégradés peuvent être simples, complexes ou fantaisistes. Il ne manque que la transparence. Mais elle arrive: Bernhard Herzog la promet pour très bientôt.

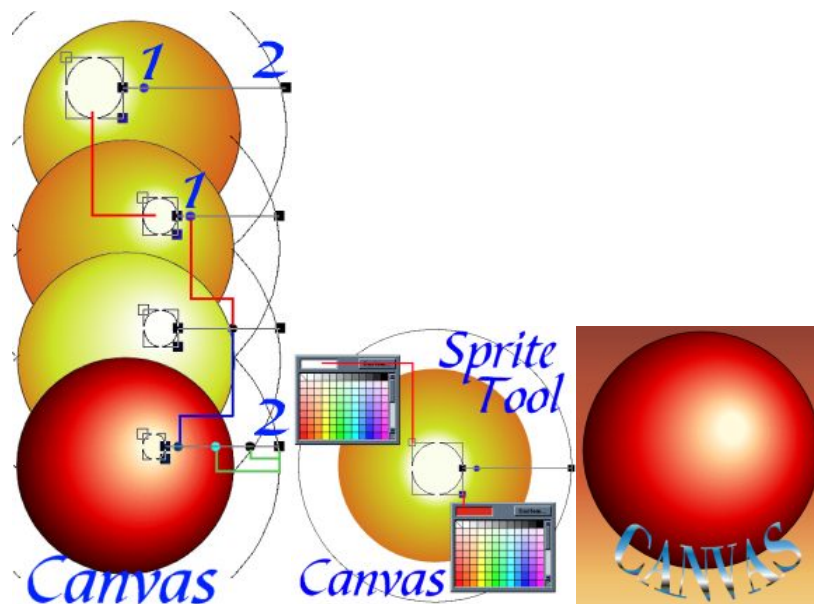


4.4- Canvas7

- 1- [Introduction](#)
- 2- [Mise au point capitale](#)
- 3- [Le pourquoi du comment](#)
- 4- [Les moyens](#)
 - 4.1 [Killustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5- [Réalisation de l'oeil](#)
- 6- [Conclusion](#)

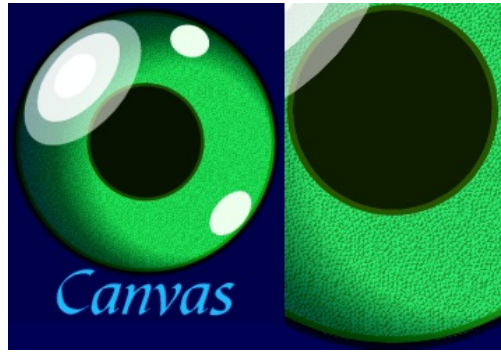
C'est l'usine à gaz, le programme typiquement Windows que l'on passe sous Linux à l'aide de Wine. On l'imagine long à se mettre en oeuvre, et l'on a raison (1 min 30 s avec un céleron 500 et 128 Mo de ram); on l'imagine lent, et l'on se trompe. Ce n'est pas un bolide, non, mais une fois chargé il devient tout à fait possible de travailler sans perdre la raison. Et l'on découvre alors une richesse de fonctionnalités inouïe. Tant en bitmap qu'en vectoriel, Canvas semble savoir tout faire. L'originalité en vectoriel vient (entre autres) d'une collection d'outils appelés Sprite Tool. Non seulement l'édition du dégradé comporte toutes options de Sketch et d'Illustrator, mais celle-ci se fait sur l'objet lui-même, et non dans une boîte de dialogue. La position du spéculaire, sa taille, la dimension des ombres, les couleurs, tout se règle à la souris et à l'aide de menus contextuels. Les réglages en sont d'autant plus rigoureux. Voilà une idée à copier sans modération, on se demande d'ailleurs pourquoi les auteurs GPL qui clonent tout avec talents n'y ont pas songé.

[Retour à la section 2D](#)



Les Sprite Tool contiennent en plus la gestion de la transparence et mieux encore: un objet se superpose à un autre avec des options de mélange identiques à celles rencontrées dans Gimp, ou Photoshop, pour le mélange des calques: addition, division, écran, brûlage que sais-je... Et si l'on ajoute qu'une forme peut se remplir avec une texture bitmap importée ou choisie dans une collection disponible à l'intérieur de Canvas, on obtient un réalisme inégalé.

Ce programme n'a que trois défauts: il tourne avec Wine, l'interface est anglaise, et c'est un produit commercial.





4.5– Artstream

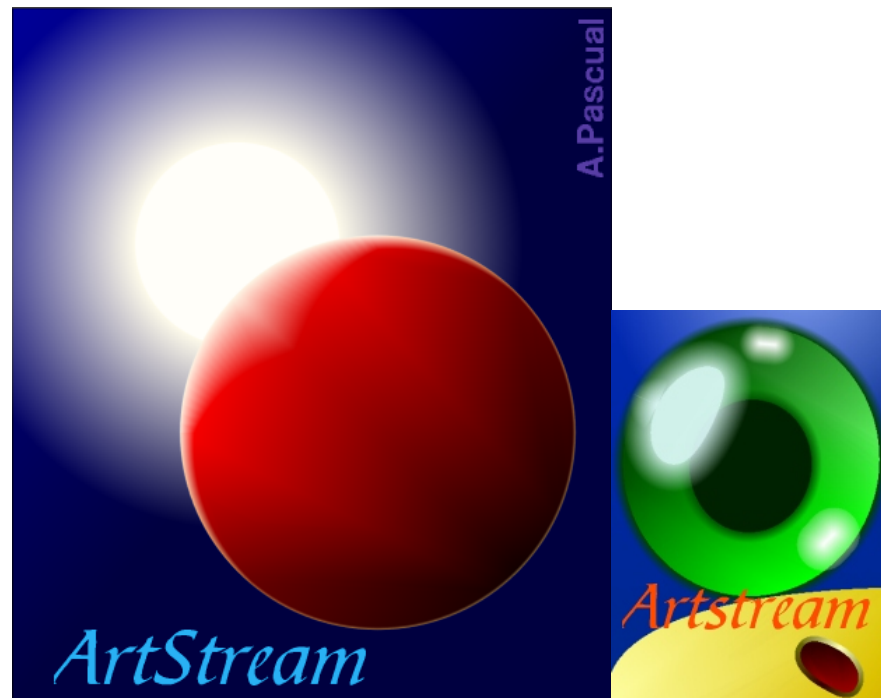
- 1– [Introduction](#)
- 2– [Mise au point capitale](#)
- 3– [Le pourquoi du comment](#)
- 4– [Les moyens](#)
 - 4.1 [Killustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5– [Réalisation de l'oeil](#)
- 6– [Conclusion](#)

[Retour à la section 2D](#)

Développé en Motif et livré en archive rpm, ce programme s'installe sans difficultés avec Kpackage. On découvre après installation une entrée Artstream dans la section Graphiques du menu K, et un nouveau type mime qui permet, lorsqu'on clique sur un fichier .asd depuis Kfm, de lancer l'application avec le fichier ouvert. C'est propre.

Initialement concocté pour SGI, Artstream est toujours en version bêta sur Linux. Des problèmes de stabilité dues à Mesa, alors qu'il a été écrit pour OpenGL, subsistent et obligent à des sauvegardes multiples.

Cela dit, ce programme ne ressemble à rien d'existant dans le domaine du dessin vectoriel et il contient des fonctions qu'il est le seul à pouvoir revendiquer. Ainsi, les dégradés simples –circulaires, linéaires...– n'existent pas pour l'instant; ils ne sauraient tarder, selon Howard Luby, son concepteur. On procédera pour ce faire par le biais de "Shade guides", analogues aux "Filets de dégradés" sur Illustrator. Très pratiques pour dégrader avec cohérence une forme tarabiscotée, ce que les autres programmes ne savent pas faire, c'est inutilement compliqué pour des formes simples. Mais toute la puissance d'Artstream se situe au niveau des points de contrôle (vertex) des courbes de Bézier. Chaque point peut recevoir une couleur et une transparence différentes en ce qui concerne l'option remplissage de la forme, et il peut recevoir une couleur, une transparence, une épaisseur et un estompage (fading) différents en ce qui concerne le contour (stroke). En insérant des points de contrôle sur le contour d'un cercle, on parvient facilement à dégrader une sphère, sans passer par les "Shade guides". Une certaine pratique est nécessaire, mais l'image des deux planètes ci-après est constituée d'un rectangle et de deux cercles, rien de plus. Tout est vectoriel, même le halo, même le spéculaire estompé sur le pourtour de la planète de premier plan. Aucune retouche en bitmap n'a été nécessaire. Qui dit mieux?



Inconvénients: interface en anglais et de facture inhabituelle, produit commercial ultra protégé: il ne fonctionne qu'avec une carte réseau installée même si l'on est pas en réseau, et après attribution d'un numéro de licence qui ne permet que d'ouvrir une session en root :-((



5- Réalisation de l'oeil

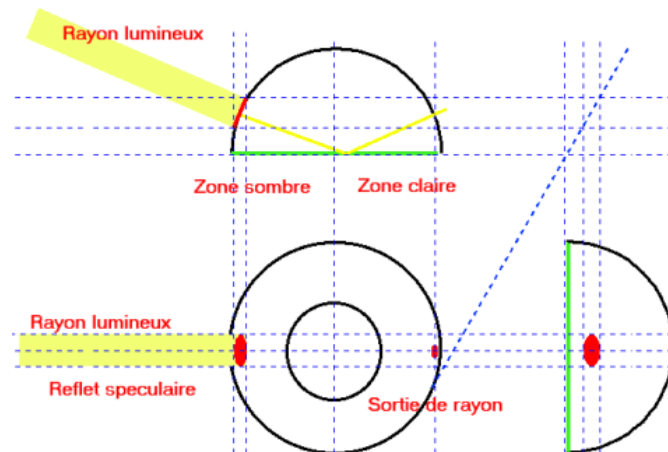
- 1- [Introduction](#)
- 2- [Mise au point capitale](#)
- 3- [Le pourquoi du comment](#)
- 4- [Les moyens](#)
 - 4.1 [Killustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5- [Réalisation de l'oeil](#)
- 6- [Conclusion](#)

Un oeil est composé pour la partie visible de la cornée translucide hémisphérique, de l'iris coloré que l'on doit percevoir et de la pupille qui n'est qu'un trou permettant le passage de la lumière vers la rétine. La difficulté est de rendre l'aspect transparent de la cornée qui par définition ne peut être vue. Cependant si l'on imagine qu'un rayon lumineux frappe une pareille surface, un reflet spéculaire doit apparaître à l'endroit où le rayon la pénètre. Poursuivant sa course, ce rayon frappe l'iris et l'illumine d'avantage, pour ressortir par effet d'incidence, en étant partiellement absorbé, produisant un spéculaire plus petit. Tout ceci n'est qu'une théorie très approximative. Mais il en découle:

-Que la partie de l'iris non illuminée par le rayon éventuel sera plus sombre que la partie recevant l'impact

-Qu'un reflet important couvrant en partie la pupille indique la provenance du rayon.

-Qu'un reflet plus petit opposé au premier indique le lieu de sortie



Eclairage de l'oeil

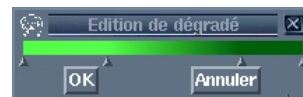
[Retour à la section 2D](#)

[Suite -->](#)

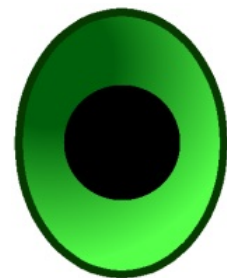


- 1- [Introduction](#)
- 2- [Mise au point capitale](#)
- 3- [Le pourquoi du comment](#)
- 4- [Les moyens](#)
 - 4.1 [Killustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5- [Réalisation de l'oeil](#)
- 6- [Conclusion](#)

Concrètement la réalisation sur Sketch (j'ai choisi ce programme parce que tout le monde ne dispose pas de Canvas ou d'Artstream) se déroule en trois étapes:
a) Créer l'iris avec un dégradé de couleur au choix, mais cohérent. On n'oublie pas de mettre une bordure

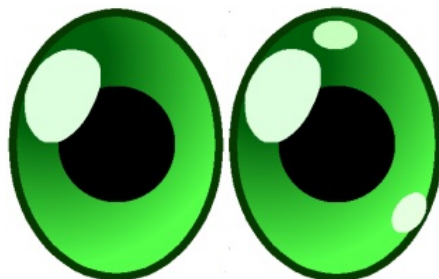


[Retour à la section 2D](#)



b) Créer la pupille: un simple cercle noir centré

c) Créer les spéculaires selon ce qui est dit précédemment. Pour faire bonne mesure, on peut ajouter un petit reflet supplémentaire censé représenter un éclat de lumière parasite.





- 1- [Introduction](#)
- 2- [Mise au point capitale](#)
- 3- [Le pourquoi du comment](#)
- 4- [Les moyens](#)
 - 4.1 [Illustrator](#)
 - 4.2 [Star Draw](#)
 - 4.3 [Sketch](#)
 - 4.4 [Canvas7](#)
 - 4.5 [Artstream](#)
- 5- [Réalisation de l'oeil](#)
- 6- [Conclusion](#)

6- Conclusion

Il apparaît qu'un rendu réaliste nécessite certaines connaissances de base physique, physiologique, et picturales.

Cela peut sembler fastidieux et décevant: on aimerait tant que tout se fasse rapidement et sans effort. N'oublions pas que Léonard de Vinci a mis plusieurs années pour achever la Joconde; Pierre Auguste Renoir également pour ses Grandes Baigneuses. Si de tels génies ont eu l'humilité de remettre sans cesse l'ouvrage sur le métier, pourquoi pas vous? Pourquoi pas moi?

André PASCUAL andre@linuxgraphic.org

[Retour à la section 2D](#)