

Sketch/Skencil_Fonte

André PASCUAL andre.pascual@linuxgraphic.org

Table des Matières

SKETCH/SKENCIL: Le caractère de la police.....	1
1– Postscript Type1 et TrueType:.....	1
2– Où trouver des fontes True Type?:.....	3
3– Convertir des True Type en Type1:.....	4
4– Quelles Fontes convertir?.....	6
5– Comment utiliser ces nouvelles fontes?.....	8
6– Premier exemple de réalisation:.....	9
7– Deuxième exemple de réalisation:.....	14
8– Troisième exemple: un peu de poésie.....	15
9– Dernier exemple: rien que des lettres!.....	17

SKETCH/SKENCIL: Le caractère de la police.

1° Partie

À l'instar de **Sodipodi/Inkscape** qui sont encore un peu jeunes (au moment de rédaction, en 2003) et des jumeaux **OpenDraw/Star Draw** qui savent tout faire et le font bien, **Sketch/Skencil** est un programme de dessin vectoriel. Il propose donc des entités mathématiques pour dessiner: *segments, polygones, cercles, courbes de Bézier...* Ces outils ont une apparence fruste et un maniement retors qui rebutent toutes bonnes volontés devant composer un travail graphique, complexe ou non. De plus, en une époque où la technicité permet de satisfaire ce déraisonnable désir d'immédiateté qui la caractérise, la pratique du dessin vectoriel, austère, quasiment ascétique, rigoureuse et nécessitant une infinie patience, s'affirme en complet déphasage avec elle. Mais, mais, mais.... La sécheresse du dessin vectoriel n'est que la surface visible des choses. Dès que l'on creuse un peu, il devient très facile de se faire plaisir avec ce type de programmes, lorsque, par exemple, en un clin d'oeil on parvient à un honnête résultat graphique par l'utilisation des seules fontes de caractères. Comment? Nous l'allons voir...

1– Postscript Type1 et TrueType:

Dans ce qui suit, nous parlerons de préférence de *fonte de caractères*, plutôt que de *police de caractères*, qui semble être un anglicisme indésirable (pléonasme!), même si le terme *police* a été conservé dans le titre du présent exposé, et qu'il pourra y ressurgir inopinément par la suite.

Une *fonte* (les caractères en plomb provenaient autrefois de chez un fondeur, d'où le nom de *fonte*) numérique est un ensemble de données susceptibles de reproduire le dessin des caractères d'imprimerie, stockées dans un ou plusieurs fichiers informatiques. Ces fichiers contiennent les informations nécessaires d'une part à l'affichage sur moniteur et d'autre part à l'impression sur un support externe de l'ensemble des caractères – *le jeu de caractères*– d'une fonte donnée.

Une fonte ne présente d'intérêt que si elle est redimensionnable, c'est-à-dire s'il est possible de l'afficher et de l'imprimer suivant des dimensions différentes. On dira alors qu'on l'utilise dans des *corps* différents. Le redimensionnement des fontes numériques, ainsi que le langage *PostScript* lié à leur gestion, sont une création de la société **Adobe**. Les fontes qui en sont issues s'appellent *fontes Type1* (il existe des Type 2 et 3), et sont caractérisées par trois fichiers descripteurs par fonte:

- *fonte.afm* (Adobe Font Metric), qui est un fichier texte
- *fonte.pfa* (PostScript Font Ascii) et *fonte.pfb* (PostScript Font Binary), ce dernier étant la version binaire du précédent, la présence simultanée des deux types n'étant pas obligatoire pour un bon fonctionnement des applications.

L'un des problèmes d'une pareille technologie vient de ce qu'il est possible d'afficher une fonte, mais non de l'imprimer, ou inversement, selon que l'on ne disposera pas de l'un ou l'autre des fichiers requis à l'une de ces tâches. L'autre difficulté vient de ce que la licence **Adobe** relative aux Type1 en rend l'utilisation chère et contraignante. Ainsi est-il prévu que les créateurs de fontes PostScript doivent verser des royalties à **Adobe**. Cela explique que sous **Linux**, qui recourait initialement aux seules fontes Type1, on disposait d'un ensemble très limité de possibilités, les fontes PostScript libres étant peu nombreuses – quelques-unes de **Adobe**, d'autres de **URW** et de **Bitstream**.

Apple, très implanté dans la PAO, dépendait donc fort étroitement de **Adobe** et de sa technologie PostScript. Voulant s'en affranchir afin d'étendre son propre contrôle sur ce domaine d'activité, **Apple** créa les *TrueType*, appelées le plus souvent **TTF** (True Type Font). Ce type de fontes vectorielles utilise un unique fichier descripteur, tant pour l'affichage que pour l'impression, qui d'ailleurs, s'effectue sur n'importe quelle imprimante de bureau, alors que les PostScript nécessitent, en principe, une imprimante du même type, ou une émulation logicielle comme **ATM**, ou **GhostScript** sous **Linux**.

La PAO professionnelle fait appel aux fontes PostScript, mais d'une manière générale, pour une utilisation courante, on pourra se fier aux fontes TrueType: le résultat imprimé sera conforme à l'affichage. De plus, il existe une très grande quantité de fontes gratuites, voire libres d'utilisation, provenant d'une multitude de créateurs indépendants ou de fondeurs de renom. Depuis quelques temps, la plupart des distributions **Linux** d'aujourd'hui gèrent les TrueType d'une façon transparente, de leur installation facile (sur **Mandrake**) à leur impression sans reproche en passant par leur affichage lissé (antialiasé). Dès lors, pourquoi s'en priver? Pourquoi ne pas en jouer graphiquement avec **Sketch/Skencil**? Après tout, un outil de dessin vectoriel doit savoir manipuler des caractères qui sont eux aussi vectoriels?

C'est ce que nous nous proposons de découvrir ici.

Note importante: les manipulations décrites dans ce qui suit, ont été élaborées sur une distribution **Linux Mandrake 8.2**. Il se peut donc que certaines informations soient à adapter pour d'autres distributions, ou d'autres versions.

2- Où trouver des fontes True Type?:

On dit que plus de 40.000 fontes de caractères ont été recensées, et qu'il s'en crée continuellement. La surabondance étant l'ennemie du mieux, il devient très difficile de trouver ce que l'on cherche exactement, d'autant que quantité de fontes se ressemblent au point que les différences échappent au regard non averti. Combien de milliers de *Times* et similaires? Combien d' *Helvetica* et analogues?... Combien de sites web dédiés aux caractères?

Sous peine de perdre son temps, il faut limiter les sources à consulter, on en conviendra. Voici donc quelques sites de très grand intérêt:

<<http://www.geocities.com/jeffsfonts/>>, de **Jeff Levine**, créateur des fontes, dont une centaine de *Dingbats*, qu'il met en libre téléchargement.

<<http://www.simplythebest.net/fonts/>>, qui semble ne pas usurper son nom, du fait de sa présentation très claire et du très grand nombre de fontes freeware ou shareware disponibles, judicieusement classées.

<<http://www.moorstation.org/typoasis/designers/westwind/>> toutes les créations, originales et soignées, de **West Wind Fonts**.

<<http://www.fontgarden.com/>>, semblable à **Simply The Best**, peut-être plus riche et différemment présenté.

<<http://www.planete-typographie.com/>>, site français, des plus instructifs qui, outre les fontes sélectionnées mises en téléchargement, offrent des pages pédagogiques concernant la typographie et l'imprimerie, qu'il s'agisse d'histoire ou de technique. A consulter absolument.

Et encore:

<<http://www.fontalicious.com/super/fonts.html>>, fontes de titrage pour la plupart.

<<http://www.fontfreak.com/>>, fontes PC et Mac.

<<http://www.1001freefonts.com/>>...

Il ne semble pas nécessaire de chercher plus loin, la plupart de sites ci-dessus proposant des liens pointant vers les créateurs et grands pourvoyeurs de freeware que sont **Ray Larabie**, **PizzaDude**, **Nick Curtis**, etc. A noter, pour l'anecdote, que **Microsoft** lui aussi propose des fontes en freeware: une seule main suffit à les compter sur les doigts. Sauf erreur de ma part, évidemment...

La plupart des fontes étant d'origine anglo-saxonne, elles ne contiennent pas de caractères accentués; beaucoup d'autres, de type fantaisie, ne contiennent que des majuscules, et seront donc réservées à la réalisation de titres. D'autres, peuvent ne contenir qu'une douzaine de caractères.

Si l'on désire disposer d'une table de caractères complète, il convient de se tourner vers les concepteurs professionnels: **Bitstream Inc**, **URW Software**, **Nimbus**, **Letraset**, **Mecanorma**, **Corel Corp.**, **Monotype Typography**, **QualiType**, **ITC**, **Adobe**... et même **Microsoft**. En effet, les fontes *Times New Roman*, *Arial*, *Verdana* et *Courier New* sont utilisées sur tous postes **Windows**: si l'on veut échanger avec eux des données provenant d'un poste **Linux**, sans altération de l'apparence, il importe d'avoir composé et mis en page lesdites données avec les mêmes caractères que ceux qui seront utilisés pour les lire. Cela va sans dire, mais cela va mieux en le disant, ainsi que l'affirme l'adage bien connu.

Si l'on a installé **Corel Photo Paint**, version 9 pour Linux, on dispose sur son système de 178 fontes True Type dont la table de caractère de chacune est complète. Inutile donc de s'en priver, et l'on pourra convertir en Type1, par exemple: *Imtpulse BT*, *Allegro BT*, *Architecture*, *Staccato222 BT*, *WP IconSymbolA*, etc. qui sont d'excellente facture.

3– Convertir des True Type en Type1:

3.1– Le nécessaire à conversion:

Rappelons un point important: les distributions modernes telles que **Mandrake** et les environnements évolués tels que **KDE** usent indifféremment des TTF ou des Types1. De ce fait, on ne convertira que les fontes nécessaires à l'usage d'un programme particulier qui ne peut recourir aux TTF. Ici, comme dit précédemment, l'opération est envisagée pour **Sketch/Skencil**, sans préjuger de son adaptabilité à d'autres programmes.

Afin de réaliser les conversions envisagées, il convient de disposer des outils adéquats, ce qui n'est guère une difficulté sous **Linux**, le choix des outils étant immense. On récupèrera, sans qu'il soit obligatoire d'y recourir dans le cas présent, les outils suivants:

–1) **ttf2pt1**, basé sur **ttf2pfa**, un autre convertisseur TTF vers Postscript. La version 3.4.3 utilisée pour cet article est en téléchargement sur:

< <http://ttf2pt1.sourceforge.net>>

Disponible depuis décembre 2002, cette version est livrée sous forme de sources compactées en une archive tgz, contenant un *Makefile* qu'il n'est donc pas nécessaire de générer par le classique *./configure*. La compilation, lancée dans le répertoire de décompression par un simple *make*, produira deux exécutables principaux: **ttf2pt1** et **t1asm**, ainsi que d'autres utilitaires de manipulation de fontes qui viennent s'ajouter à la douzaine de scripts livrés avec l'archive. De cet ensemble, seul **ttf2pt1** nous intéressera par la suite.

–2) **type1inst**, de **James Macnicol**. Il s'agit en réalité de deux scripts en Perl, **type1inst** et **t1embed**, disponible en archive tgz sur:

< rpmfind.net/linux/RPM/type1inst.html>.

Ces outils servent à générer dans le repertoire des fontes installées les fichiers descripteurs: *fonts.scale*, *fonts.dir* et *Fontmap*. Dans le cas de **Sketch/Skencil**, il est préférable de recourir à l'utilitaire maison pour générer *fonts.scale*, qui différera de celui produit par **type1inst**, lorsque certaines fontes converties présentent des difficultés. Au final, **type1inst** s'avère utile uniquement pour générer *Fontmap*, fichier nécessaire à **Ghostscript**, mais inutile pour notre démonstration.

–3) **Type1Utils**, de **Eddie Kohler**. Livré en sources tgz, ce paquetage produit six exécutables après compilation: **t1ascii**, **t1asm**, **t1binary**, **t1disasm**, **t1mac**, **t1unmac**, destinés à convertir des *PFB* en *PFA* et inversement, à désassembler ou assembler ces mêmes fichiers, et à convertir les *ATM/Laserwriter* de **Macintosh** en Type1. Non utilisée pour le problème qui nous occupe, cette collection d'utilitaires est signalée à toutes fins utiles. A télécharger sur:

< <http://www.lcdf.org/~eddielwo/type/t1utils-1.26.tar.gz>>

–4) **mkfontdb.py**, script Python de **Bernhard Herzog**. Est indispensable pour générer un *fonts.scale* correct et *std.sfd*, c'est-à-dire *standard sketch font directory*, à télécharger sur le site officiel de **Sketch/Skencil**:

< <http://sketch.sourceforge.net/mkfontdb-1.3.tar.gz>>

3.2– Exemple de conversion:

Pour une plus grande clarté de lecture, créons un répertoire temporaire dans son *home*, par exemple *Convfont*. Copions-y une fonte *TTF*, n'importe laquelle, par exemple la très belle *staccato.ttf*, du fondeur **Bitstream Inc**.

En mode console, si l'on a proprement installé **ttf2pt1**, il suffira de se rendre dans le répertoire *Convfont* contenant la fonte à convertir et d'entrer la commande:

ttf2pt1 -ab staccato.ttf

La mention **-ab**, en réalité les options **a** et **b** du programme, signifie que l'on désire une table de caractères contenant tous les caractères inclus dans le fichier *TTF* d'origine (option **a**) et que celle-ci doit être assemblée dans un fichier compressé binaire *pfb* (option **b**).

A l'exécution de la commande, des indications vont défiler, notamment l'annonce de la création d'un

fichier nommé *staccato.pfb*; on apprendra aussi que la fonte convertie s'appelle très précisément: *Staccato222BT-Regular* et que l'opération de conversion sera terminée à l'apparition du message suivant: *Finished. Font File Created*. L'examen du répertoire *Convfont* montrera alors que deux fichiers viennent d'être créés: *staccato.afm* et *staccato.pfb*.

Le fichier *afm* étant un fichier texte, il est facile de le lire pour y apprendre que la table de caractères contient 307 signes (*StartCharMetrics*), que le cé cédille minuscule (ç) est le caractère 184 dans la table, alors que le même en capitale est le caractère 199 (Ç) etc.. Si l'option `-a` n'avait pas été demandée lors de la conversion, la table de caractères n'en contiendrait que 256. A chacun de voir s'il veut disposer des signes comme la tilde (~), le dollar (\$), dièse (#) etc...

A noter que la liste complète des options de **ttf2pt1** s'obtient en entrant simplement dans une console: **ttf2pt1**. Pour clore sur le sujet, ajoutons que l'appel du programme sans option limite le résultat de la conversion à un fichier *staccato.afm*, et que l'option `-e` produit un fichier *staccato.pfa*. A chacun de voir s'il préfère les *.pfa* aux *.pfb*.

3.3- Exemple d'installation:

L'installation de la fonte (ou des fontes, évidemment) consiste à la placer dans le répertoire idoine, puis à la faire prendre en charge par **XFree** et par **Sketch/Skencil**. Cela peut paraître compliqué, surtout comparé à la simplicité de procédure requise par **Windows** pour installer une fonte, mais ce ne l'est pas vraiment: il suffit d'un minimum de rigueur pour mettre en oeuvre ce qui suit.

1) Tout d'abord, copier les fichiers *staccato.afm* et *staccato.pfb* dans:

/usr/lib/sketch-0.6.15/Ressources/Fontmetrics.

L'opération devra être menée dans un terminal en mode SU, soit en ligne de commande (**cp [options] source cible**) pour les masochistes, soit en appelant l'un des irremplaçables **MC, KruSader ou Konqueror**.

2) Entrer dans le répertoire */usr/lib/sketch-0.6.15/Ressources/Fontmetrics* (**cd [-L|-P] [dir]**).

3) Installer la nouvelle fonte en appelant soit:

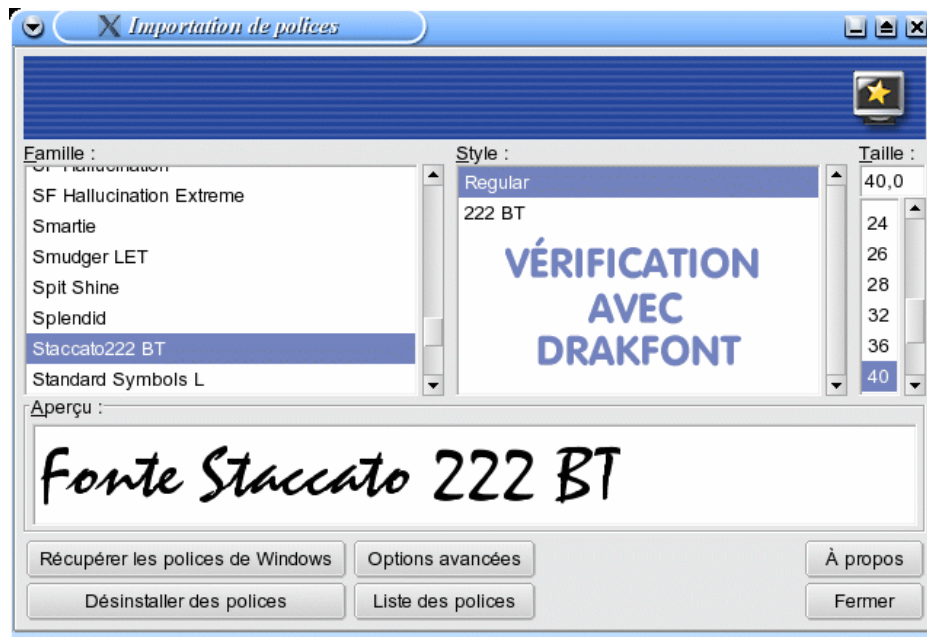
`-type1inst` sans options, ce qui, après scrutation du répertoire *./Fontmetrics*, générera les fichiers *fonts.dir*, *fonts.scale* et *Fontmap*. ce dernier n'ayant aucune utilité si son contenu n'est pas rapporté dans le *Fontmap* de **Ghostscript**. Toutefois, il est apparu à l'usage que les fichiers *fonts.dir* et *fonts.scale* ne sont pas toujours corrects pour une utilisation sans histoire par **Sketch/Skencil**. On recourra alors à la solution suivante, qui est valide à coup sûr:

`-mkfontdb.py -x -s`, qui générera les fichiers *fonts.scale* et *std.sfd* en adéquation avec ce que **Sketch/Skencil** attend. Si l'on tient à avoir *fonts.dir*, il suffira de copier *fonts.scale* en le nommant *fonts.dir*, ces deux fichiers contenant en principe les mêmes informations.

4) Demander à **XFree** de prendre en compte les fontes de **Sketch/Skencil**, ce que l'installation de **Sketch/Skencil** ne fait pas par défaut, et qui devrait être réalisé avant la première utilisation de **Sketch/Skencil**. Pour cela, éditer le fichier */etc/X11/fs/config*, ajouter une virgule en fin de la dernière énumération des répertoires de fontes de la section *catalogue*, et ajouter la ligne: */usr/lib/sketch-0.6.15/Ressources/Fontmetrics*, sans mettre de virgule en fin de ligne. A noter que l'opération pourrait être réalisée grâce à **/usr/sbin/chkfontpath [options]**

5) Faire prendre en compte l'ajout de la nouvelle fonte par le système, soit de la façon brutale, mais brutalement efficace, consistant à faire redémarrer son système soit , toujours dans un terminal en mode SU, en entrant la commande: **xset fp rehash** (consulter la documentation à ce propos).

6) Vérifier –c'est facultatif– que l'installation est correcte en appelant **K Char Select**, ou **Drakfont**. (Voir illustration).



4– Quelles Fontes convertir?

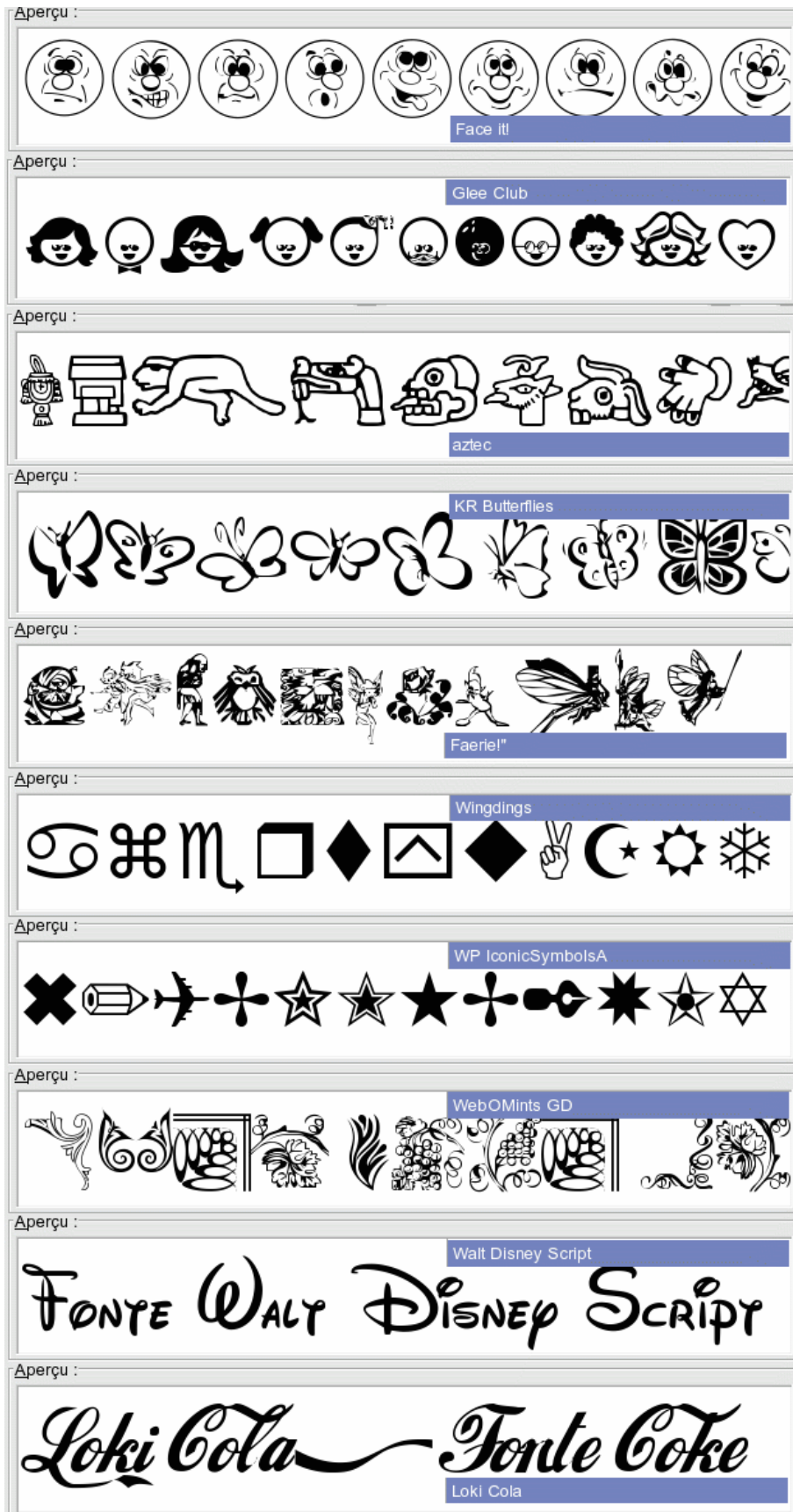
Evidemment, il n'y a pas de réponse absolue. Chacun opèrera en fonction de ses goûts, de ses besoins et de ses coups de coeur. Cependant, il existe des fontes dont les caractères sont des glyphes –terme parfaitement adéquat pour les fontes *Aztec* ou *Mayan*– non alphabétiques, couvrant bien des domaines ainsi graphiquement représentés. Il serait alors peu judicieux pour un graphiste de s'en passer tant il est possible, en les utilisant, de composer rapidement une illustration comme il a été dit en introduction. Les fontes les plus représentatives du genre sont les *Dingbats*; elles sont devenues une référence, au point que toute fonte composée d'autre chose que de caractères alphabétiques est classée usuellement dans les catégories des *Dingbats*, quel que soit le domaine qu'elle couvre.

Ainsi, on trouvera:

- des symboles divers: *Dingbats*, *Wingdings*, *WP IconicSymbolsA*, *Webdings*, *2 TheLeft Dingbats*, *PizzaDude Bullets*...
- des flèches: *Arrows*...
- des encadrements floraux: *Corners*, *Flowers Show*, *WebOMint*...
- des personnages: *Gee Club*, *Tombats Four*, *Mr Men*, *McZee*, *FlyingPenguin*, *FFDingBest*, *Face It!*, *Face On!*, *Dont bug Me*....

–des hiéroglyphes en tout genre: *Mayan, Aztec, PharaoGlyph, Deities...*

Des mammifères, des insectes, tout ce que l'on voudra, et même des fontes qui imitent les caractères propres à certaines raisons sociales: *Loki Cola, Walt Disney Script...* Voir illustration de spécimen.



5– Comment utiliser ces nouvelles fontes?

Une fonte de caractères, on l'aura compris, est une collection de petits dessins –les glyphes– représentant des lettres ou n'importe quoi d'autre. Chaque caractère est constitué de courbes de Bézier: il s'agit donc d'une entité vectorielle mathématiquement définie qui pourra être redimensionnée, déformée, triturée point par point. C'est en ce sens que les fontes de caractères sont de précieux alliés des graphistes: du fait de leur malléabilité. Cependant, il ne sera possible de les plier à volonté que sous certaines conditions.

Au départ, dans tout programme de dessin vectoriel, une chaîne de caractères est une entité de texte, et elle le demeurera tant qu'on ne la transformera pas en autre chose. Si l'on envoie à un imprimeur un dessin produit par **Sketch/Skencil** contenant du texte en tant que tel, et que l'imprimeur ne dispose pas de la fonte utilisée pour composer ce texte, le dessin ne pourra pas être ouvert, ou bien la fonte sera remplacée par une fonte de substitution qui ne garantira pas la mise en page originale.

*NOTA :on traite avec l'imprimeur sur la base de dessin **Sketch/Skencil** converti en .ai, tous les imprimeurs disposant de **Illustrator** , et aucun de **Sketch/Skencil***

Pour éviter les obstacles de ce genre, il suffit que le dessin ne contienne pas de texte mais uniquement des entités de dessin. A cette fin, les chaînes de caractères seront converties en dessin par la commande: **>Courbe > Convertir en courbe**. Cette commande, qui existe dans d'autres programmes, peut s'appeler différemment.

Si l'on veut manipuler séparément chaque caractère (qui a perdu son attribut de texte), il faut le rendre indépendant des autres par la commande: **>Disposer >Dégrouper**.

Si l'on veut modifier le dessin du pseudo-caractère, il faut se placer en **Mode d'édition** de points.

Cette procédure primordiale se trouve résumée dans l'illustration suivante.



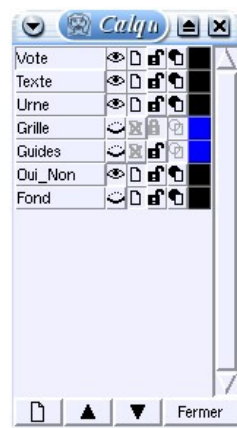
6– Premier exemple de réalisation:

Proposons–nous de réaliser une affichette incitant à l'accomplissement du devoir civique par excellence: voter lors d'un référendum, où il ne s'agit que de déposer dans l'urne un bulletin: Oui, ou un bulletin: NON, selon les convictions de chacun. L'affichette en question pourrait ressembler à l'image suivante, toute considération esthétique étant laissée de côté. Il n'est ici question que de technique.

*NOTA: il est entendu que pour ce genre d'exercice, la manipulation minimale de **Sketch/Skencil** est prérequis.*



L'habitude de l'illustration vectorielle dit rapidement que cette image nécessite différents calques (couches) correspondant chacun aux éléments qui la composent: un fond, une urne, l'action de voter (main et bulletin), du texte standard et le contenu possible du bulletin (oui, non) variable selon l'opinion du votant. La structure en calques de l'illustration pourrait donc ressembler à ceci:



6.1– Création du Oui_Non:

On peut supposer que le "Oui" est joyeux, et qu'une fonte enjouée devrait le traduire. La fonte *Walt Disney Script* (<http://members.aol.com/disneyfonts/wds.html>) conviendra. Quant au "NON", qui veut s'opposer violemment, une fonte massive comme l'*Impact* de **Monotype Corporation** devrait en montrer la détermination. Evidemment, comme en tout scrutin, il y a les hésitants, qui naviguent entre les deux possibilités. Transcrire cela graphiquement est fort possible avec **Sketch/Skencil**. Quatre étapes seront nécessaires à la réalisation de la chose.

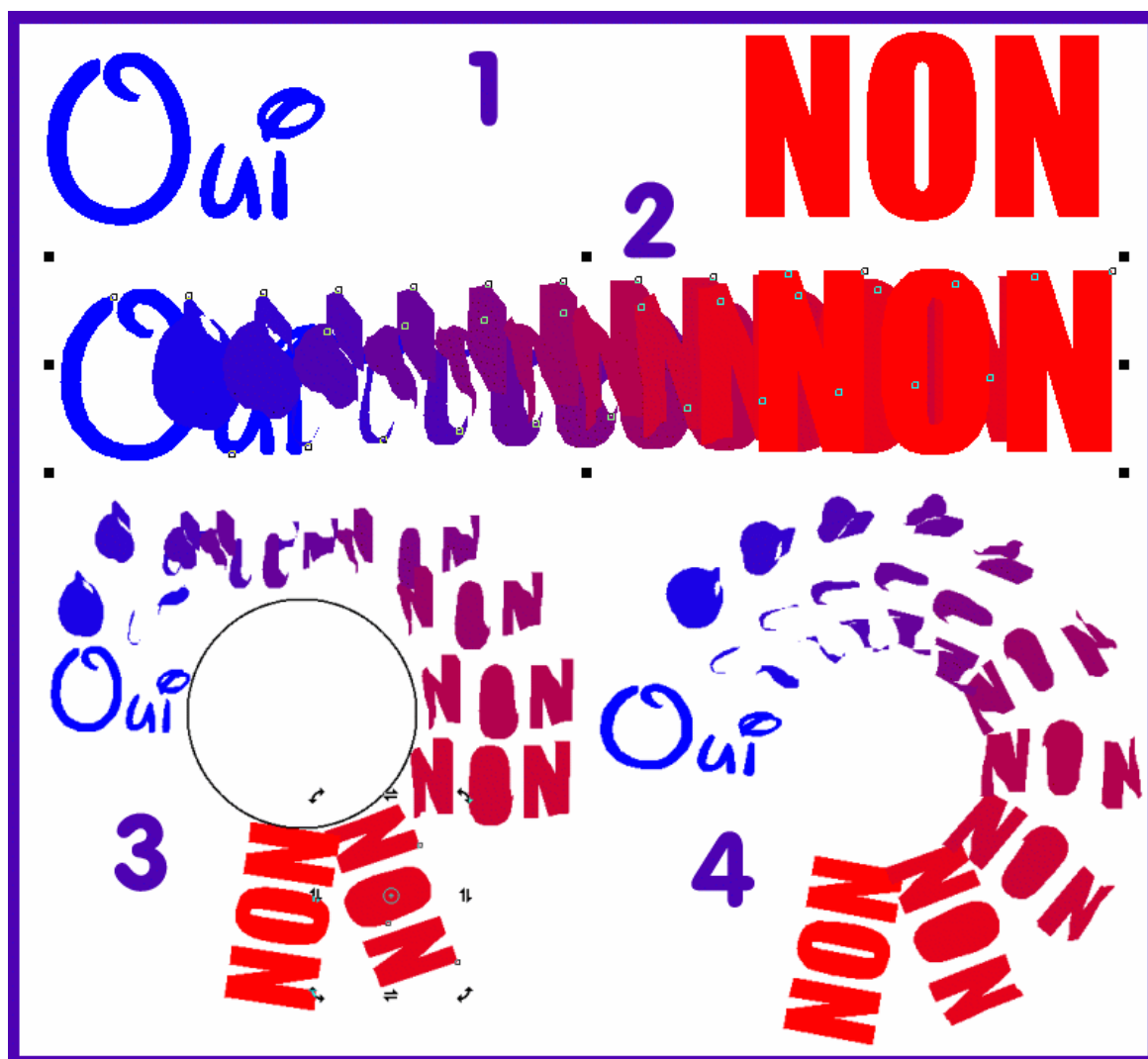
1° Etape: sur le calque *Oui_Non*, entrez le texte "Oui", de couleur bleue, puis le texte "NON" de couleur rouge, chacun des deux textes dans la fonte choisie. Convertir les deux textes en courbes.

2° Etape: sélectionner les deux textes (qui sont maintenant des entités de dessin), puis dans **>Effets >Transformations (ou mélanges)> Etapes:10 >Appliquer**. Un *morphing* (transformations successives, appelées *Dégradé de forme* dans **Illustrator**) de forme et de couleur est alors réalisé entre le "Oui" et le "NON".

3° Etape: dégroupier le *morphing*, tracer un cercle et disposer radialement (et à la main!) chaque élément de la transformation autour du cercle. Rappelons qu'un clic sur un objet le sélectionne et autorise son déplacement, alors qu'un second clic permet sa rotation et son itilisation.

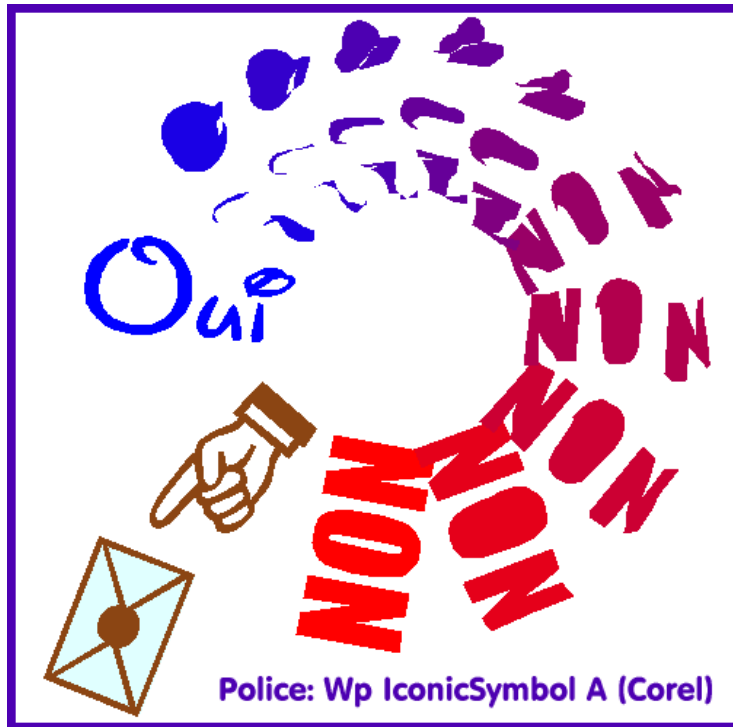
4° Etape: supprimer le cercle qui n'avait qu'un rôle de guide.

Voir illustration des quatre étapes.



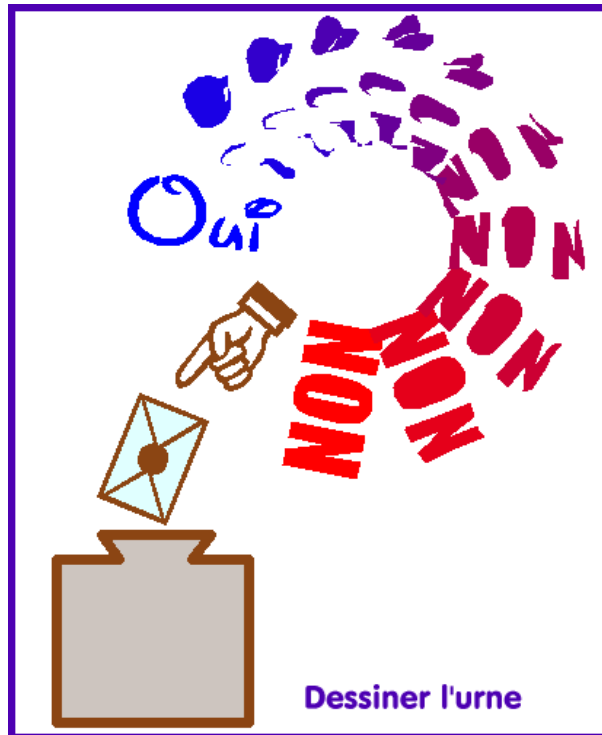
6.2– Création de l'action de voter:

Plus explicites qu'un discours, quelques pictogrammes diront clairement que voter consiste à déposer une enveloppe fermée contenant son bulletin, dans une urne. La fonte *WP IconicSymbolA* de **Corel** contient les deux éléments dont nous avons besoin: une enveloppe cachetée et un main, index tendu, indiquant une direction à suivre. Rappelons que l'on peut connaître le contenu d'une table de caractères, ainsi que la touche du clavier à laquelle un caractère correspond grâce, par exemple, à **Gnome Character Map**. Ici, la main correspond à la touche: "L" et l'enveloppe correspond à la touche "J". Bien évidemment, ces caractères seront convertis en courbes et dégroupés avant mise en place sur le calque *Vote*. Voir illustration.



6.3– Dessin de l'urne:

Celle-ci n'étant pas disponible comme caractère (peut-être existe-t-elle dans une autre fonte), la dessiner sur le calque *Urne* grâce à l'outil: **Dessiner Polyligne**. Ne pas oublier d'éditer le tracé en mode points (de contrôle), de sélectionner les points d'origine et de fin de tracé, et de clore la figure grâce à **>Courbe >Joindre noeuds**. On pourra ainsi remplir la figure sans qu'il y ait de fuite. Voir illustration.



6.4 Ajouter du texte:

Sur le calque *Texte*, ajouter en **VGA Rounded** de **Bitstream Inc**, les textes "ou" et "...mais VOTER!". Convertis en courbes, ces textes pourront recevoir une couleur de remplissage et une couleur de contour (ainsi qu'une épaisseur de contour) différentes. Voir illustration.

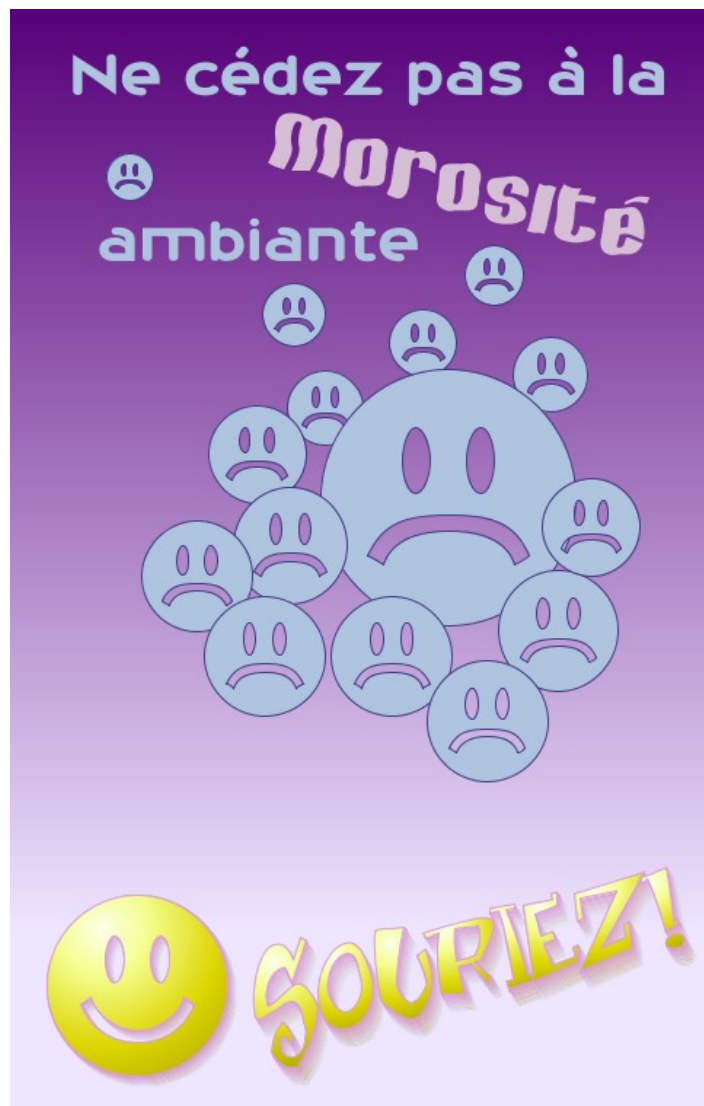


6.5– Ajouter un fond:

Sur le calque *Fond*, tracer un rectangle aux dimensions de la page A4 et le remplir d'un dégradé, d'un aplat ou d'un motif, selon les préférences de chacun. Ne reste plus alors qu'à signer l'oeuvre que l'on s'était proposée de réaliser en début d'exercice. Facile, non?

7- Deuxième exemple de réalisation:

Cet exercice est fort simple à réaliser, puisqu'il ne contient que du texte converti en courbes. Il est inutile donc de le trop détailler. Les fontes de caractères ont été choisies en fonction du message que chaque texte exprime. Ainsi, "Morosité" est d'un dessin flasque et lourd à la fois, dans une position descendante comme on peut s'attendre d'un sentiment qui achemine vers la dépression, et dans une couleur fade et froide. Il s'agit de la fonte *SF Hallucination*, de **ShyFonts Type Foundry**. Le texte "SOURIEZ!", au contraire se doit d'avoir une allure guillerette, ascendante, pleine de lumière, de chaleur et de volume. Il s'agit de la fonte *Junior & Stinky*, de **John Martz**. Quant au texte principal, donneur de conseil, il se doit d'être docte, lisible, sans fantaisie, droit comme une rembarde. Il s'agit de la fonte *20th Century Font*, de **Ray Larabie**. Restent les personnages: deux caractères provenant de la fonte *PizzaDude Bullets*, de **Jakob Fischer**. Le personnage morose est obtenu grâce à la touche "5" du pavé numérique, et le personnage souriant grâce à la touche "3". Les grises mines reçoivent un remplissage en aplat, alors que la mine réjouie est remplie avec un dégradé circulaire qui lui donne du volume. De plus, lui-même et le texte "SOURIEZ!" qui l'accompagne, rempli de la même façon, sont dupliqués sur un calque sous-jacent afin de créer une ombre légère qui leur donne ainsi du relief, en contraste avec la platitude des éléments moroses. Le tout est disposé sur un fond dont le dégradé va du sombre au clair, pour exprimer le passage du triste au joyeux. Voir illustration. Simplissime, non?



8- Troisième exemple: un peu de poésie.

La famille des *Dingbats* couvre, comme il a déjà été dit, un grand nombre de domaines. Si la poésie (naïve) évoque les fleurs, les fées, les papillons... il est alors aisé de traduire graphiquement ce concept grâce aux fontes disponibles. En voici une interprétation possible: "...et volent les papillons".



Sur cet exemple encore, pas un seul trait n'a été tracé: il ne s'y trouve que des caractères. Les motifs floraux proviennent de la fonte *Corners 1*, de *Helen Duggan*, les papillons de la fonte *KR Butterflies* (<http://www.katsfunfonts.com>) et le texte de la fonte *Mandingo*, de *Font "a" Licious Fonts*. L'illustration

se compose au minimum de trois calques: *Fond*, *Ombre* et *Motifs*, en allant du plus bas vers le plus haut. Les motifs (les caractères convertis en courbes) sont remplis par des aplats, avec des bordures ayant leur tonalité accordée au remplissage. Les bordures ont pour but de souligner les formes, qui cependant restent assez plates sur un fond vide: élément (1) sur l'illustration de la décomposition en calques. Pour donner du relief à l'ensemble sans distraire le regard, il convient de créer sur le calque *Fond*, un arrière-plan uni de couleur claire (2). Mais l'image demeure encore plate. Aussi aura-t-on tout avantage à sélectionner le contenu du calque *Motif*, et à le copier dans le presse-papier pour le coller ensuite sur le calque *Ombre*, où la copie recevra une couleur unie identique pour le remplissage et les bordures. Cette couleur sera semblable à celle du fond, mais en plus sombre (3). Le contenu du calque *Ombre* sera légèrement décalé par rapport au contenu du calque *Motifs* afin de créer l'effet d'ombre recherché (4). Élémentaire, non?



9- Dernier exemple: rien que des lettres!

Il n'est pas un logiciel de dessin vectoriel qui, dans l'exposé de ses fonctions, ne présente pas la possibilité de réaliser une image avec seulement des chaînes de caractères (alphabétiques). **Illustrator** n'échappe pas à la règle. Il est donc juste de montrer que **Sketch/Skencil** en est également capable. L'illustration suivante le prouve et nous verrons dans un prochain numéro comment parvenir à ce résultat sans souffrir le martyre. La procédure sera explicitée dans le détail et constituera ce que nos cousins du Québec n'appelleraient pour rien au monde un "Step by step".



Derniers babillages: si l'on préfère les machineries lourdes de possibilités multiples, on se tournera vers **StarDraw/OpenDraw**. Tout ce qui vient d'être dit plus avant pourra y être réalisé de manière semblable avec, en plus, l'introduction de fonctionnalités qui font cruellement défaut à **Sketch/Skencil**: la transparence et le dégradé de transparence, tant sur les zones de remplissage que sur les contours. Et ne parlons pas de cet outil hautement créatif qu'est la conversion 3D de tout objet, bitmap ou vectoriel, et de son texturage, ni de l'avantage à utiliser dans **StarDraw/OpenDraw** les fontes **TTF** sans transformation préalable en **PostScript Type1.... et c'est tout pour aujourd'hui!**

NB : On trouvera la 2^o partie de cet exposé sur [Linuxgraphic](#), dans la même section, au format PDF et sous le nom de : [Sketch/Skencil fait le masque](#).

André PASCUAL
andre.pascual@linuxgraphic.org

Cet article, rédigé en mai 2003, a été publié dans **LinuxPratique France** en 2004. Le présent PDF a été généré avec HTMLDoc le 15 février 2005, pour être mis en ligne dans les jours qui ont suivi.