

Merci à ma classe de DNSEP Design Graphique pour les moments partagés, à toute l'équipe enseignante pour leur dévouement et leur suivi, à Jacques André, Alice Savoie, Yannick Mathey, Louis-Rémi Babé et aux forumeurs de Typedrawers pour m'avoir éclairé de leurs lumières expertes. Merci à Samuel Gadea, Amélie Guesnon et Clara Chague pour leur soutien inconditionnel.

2016, Mémoire de DSNEP, Baptiste Guesnon
ÉSAD. Grenoble—Valence

Caractères versatiles

En quoi notre usage de la typographie
réactualise-t-il la question de son dynamisme ?
Pourquoi n'y sommes-nous pas déjà,
et comment y parvenir ?

Introduction

P 22

P 22

P 27

P 37

P 45

P 51

P 51

P 63

P 69

P 79

P 85

P 93

P 103

P 103

P 109

P 115

P 121

P 129

- 1 Espace fixe, espace mouvant
 - A Dynamique/Statique
 - B Dynamique du format et de la lettre
 - C Définir le dessin par le programme
 - D Texte en mouvement

- 2 Espace formaté
 - A Le texte dans l'espace
 - B Le format typographique
 - C Décomposer les glyphes en modules
 - D Décrire la lettre par ses contours
 - E Définition et résolution d'affichage
 - F Un dessin multiple

- 3 Évoluer dans l'espace
 - A Vers quelle typographie dynamique ?
 - B Dépendance aux interfaces utilisateurs
 - C Construire des outils sur mesure
 - D Les enjeux de la typographie dynamique
 - E Qui laisser intervenir sur la forme du texte ?

Ressources

| | | |
|-------|---------------|-------|
| P 143 | Bibliographie | P 185 |
| P 147 | Webographie | P 185 |
| P 153 | Vidéographie | P 193 |
| P 157 | Citations | P 209 |
| P 165 | Lexique | P 217 |
| P 169 | Biographies | |
| P 173 | Annexe | P 225 |
| | | P 225 |
| | | P 233 |
| | | P 243 |
| | | P 251 |
| | | P 251 |
| | | P 263 |
| | | P 275 |
| | | P 289 |
| | | P 295 |
| | | P 295 |
| | | P 303 |
| | | P 317 |
| | | P 317 |
| | | P 323 |
| | | P 333 |
| | | P 333 |
| | | P 346 |

A Évolution de la composition

- [1 — 6] Composition au plomb
- [7 — 19] Composition mécanique
- [20 — 24] Photocomposition
- [21 — 29] Composition numérique

B Les dispositifs d'affichage

- [1 — 5] Impression et papier
- [6 — 13] Écrans
- [14 — 18] Échelles de lecture

C Création de caractères

- [1 — 10] Rationnalisation
et géométrisation
- [11 — 19] Description de la lettre
- [20 — 31] Influence formelle
de la technique
- [32 — 36] Hinting

D Les outils du créateur typographique

- [1 — 5] Outils analogiques
- [6 — 17] Outils numériques

E Famille de caractères

- [1 — 4] Famille étendue statique
- [5 — 11] Fontes dynamiques

F Relation typographie/contexte

- [1—12] Contrainte de l'espace
- [13—15] Relation forme/sens

Introduction

Le contexte dans lequel évolue la typographie, telle que nous la connaissons aujourd'hui, n'a plus grand-chose à voir avec celui dans lequel elle a été inventée. Aujourd'hui, le texte est mobile, changeant. De la *Une* des journaux à la bourse en passant par les e-mails, les panneaux indiquant les heures de passage des moyens de transport... les contenus et les formes sont en perpétuelle métamorphose. Pourtant la typographie que nous utilisons, bien que numérique, est pensée comme un matériau statique. La typographie numérique est pourtant parvenue à une certaine maturité. Nous avons aujourd'hui accès à des formats logiciels typographiques offrant la possibilité de substituer des glyphes et permettant des rendus de haute qualité graphique des caractères. En effet, avec la création du format *Opentype* [ANN. 19], les formats de description et de gestion des typographies sont unifiés et utilisables sur tous les supports et systèmes d'exploitation, homogénéisant les formes typographiques, quels que soient les supports. Cependant, le passage d'une technologie mécanique [IMG A, 7 – 19] à une typographie digitale [IMG A, 20 – 24] n'a pas encore tenu toutes ses promesses. Si le formatage du texte peut désormais être produit par des algorithmes, permettant ainsi la résolution de problèmes ancestraux tels que celui de déterminer le moment où une ligne se finit ou celui des espaces entre les mots et les caractères dans le cas d'un paragraphe justifié ou encore les césures des phrases et des mots, l'outil informatique a davantage complexifié la typographie qu'il ne l'a sim-

plifié. Téléphones, montres, tablettes de toutes tailles, écrans d'ordinateur, tableaux d'affichage... [IMG B, 15 – 19] Chacun de ces dispositifs implique une échelle et un mode de lecture propre. Chacune de ces situations de lecture correspond à un rapport spatial et corporel différent auquel le texte et donc le dessin typographique doit s'adapter. La typographie a déjà tenté de répondre, avec plus ou moins de succès, à ces questions dans le contexte du support imprimé. Aujourd'hui le contexte numérique les actualise. À l'utilisation de différents corps optiques [LEX13], nous préférons la simple mise à l'échelle [LEX14]. À l'encrage manuel de nos caractères sur la presse, nous avons laissé place au moteur de rendu de nos imprimantes. La typographie n'a jamais été aussi parfaite. Pourtant pouvons-nous en dire autant du design ? Et si nous avons perdu quelque chose en chemin ?

Nous avons aujourd'hui la capacité technique d'atteindre une qualité jusqu'alors inégalée et de créer les impressions les plus léchées que l'on puisse imaginer. Malheureusement, les résultats ont trop souvent été parfaitement inintéressants. La qualité d'un produit imprimé, la précision de son réglage typographique, la perfection de son impression ne sont pas nécessairement ce qui fait un bon design ou une communication claire. [w25] [C6]

L'enjeu d'une typographie au dessin dynamique ne se trouve probablement pas seulement dans la possibilité d'une typographie plus précise optiquement, plus efficace techniquement ou plus fine visuellement. Traditionnellement, notre interaction avec notre environ-



nement textuel est régie par les designers ou les maquetistes qui opèrent des choix pratiques, optiques et esthétiques. Les choix concernant la typographie elle-même sont opérés par le designer typographique qui décide de la couleur, du dessin, du rythme du caractère. Que se passerait-il si les designers disposaient d'une plus grande maîtrise typographique, s'ils pouvaient définir des comportements typographiques plutôt que de se restreindre à des choix esthétiques figés ? Qu'induirait une typographie qui s'adapte à l'espace dans lequel elle évolue ou à l'échelle à laquelle on l'utilise ? Quels seraient les principes de décision permettant de faire varier le dessin des glyphes et à qui seraient-ils destinés ? Une typographie aux dessins variable engendrerait une remise en question de la gestion du rythme, du gris du texte ou encore de sa mise en forme. À l'origine de ces questions, on trouve une idée simple : et si les formes typographiques avaient un dessin variable ? Ma recherche s'appuiera sur une interrogation des relations entre les techniques et les formes et c'est à partir de mon regard de praticien que je me propose de la mener.

Ce mémoire étant rédigé par un designer et non un informaticien, je ne m'attarderai qu'à minima sur les aspects techniques sous-tendus par l'ingénierie logicielle des fontes. La compréhension technique de ce qu'est une fonte n'est pas indispensable pour apprécier ce que signifie une typographie au dessin variable. Il me semble néanmoins important de prendre le temps de s'y pencher afin de ne pas rester au stade hypothétique, voire fantasmé, de ce que pourrait être une fonte dynamique. Il serait de plus dommage de se priver de ce que la courte

— mais riche — histoire de la typographie digitale peut nous apprendre sur le sujet. Une annexe placée à la fin du texte permet au lecteur qui le souhaite d’approfondir le sujet. Je ferai référence à celle-ci régulièrement sans que sa lecture en soit pour autant nécessaire. L’écriture de ce mémoire m’a mené à enquêter sur le fonctionnement logiciel de la typographie et mieux connaître son histoire, je souhaite que cette annexe en constitue un résumé utile.

- 1 Espace fixe, espace mouvant
 - A Dynamique/Statique

Il me semble nécessaire avant toute chose d'éclaircir un point de vocabulaire. On trouve assez rapidement le terme de « dynamique » pour qualifier les expérimentations et réflexions sur la typographie allant dans le sens d'une variabilité et d'une adaptation du dessin des lettres à leur contexte [B8]. Mais, au-delà du consensus qui semble s'être dessiné autour de cette appellation, apparaît une ambiguïté. En effet, le terme dynamique évoque un mouvement, une évolution dans le temps. Or, bien que la typographie dont nous parlons puisse varier au cours du temps, et apparaître en mouvement, le mouvement n'est ici qu'une conséquence de la variation du dessin, et non un but en soi. J'ai demandé à Jacques André, l'un des pionniers des expérimentations typographiques dynamiques dans les années 1980, pourquoi il utilisait ce terme « dynamique ».

Voici sa réponse :

En informatique, dans les langages de programmation, on utilise toujours le double concept (ou l'antinomie) statique/dynamique. Pour prendre une image naïve, si vous avez encore des souvenirs de vos mathématiques de quatrième ou seconde, quand vous écrivez :

$$2x^2 - 6x + 4 = 0$$

il n'existe qu'une seule solution (avec les deux valeurs $x=1$ et $x=2$). Si vous donniez cette équation à un dessinateur de courbes (comme ceux utilisés dans les fontes), vous auriez une et une seule parabole.

Si vous écrivez maintenant :

$$ax^2-6x+4=0$$

Selon la valeur de « a », alors vous obtenez une infinité de paraboles, disons emboîtées, et si vous choisissez une valeur précise de « a », vous n'en choisissez qu'une. Le mot dynamique est bien en relation avec le temps, mais pas avec la durée, mais plutôt avec l'instant où on fait les choses, c'est-à-dire pour nous, quand on fait le calcul. Pour les fontes normales, on calcule les courbes de Bézier (c'est-à-dire le positionnement exact des contours dans la matrice de point) une fois pour toutes dans un texte (pour un corps donné, un angle donné, etc.), à la première occurrence. Avec une fonte dynamique, les courbes sont calculées « dynamiquement » (au sens informatique : au moment de l'appel). Chaque fois qu'on « appelle » un « a », on recalcule ses courbes.

Ily a quand même mouvement, car ça ressemble à ces photos de danseurs prises à intervalles très courts et superposées où le mouvement est marqué par le déplacement des bras par exemple. Mais si on regarde une seule photo, pas de mouvement. [1]

Il faut donc comprendre l'usage du terme dynamique en référence à sa définition informatique et mathématique, en lien avec son antonyme statique. Si ce terme ne me satisfait pas complètement, j'ai rapidement été confronté au fait qu'il était compris et utilisé par tous mes interlocuteurs spécialistes de la typographie digitale. Ainsi je parlerai au cours de ce texte par alternance de

[1] Extrait d'un échange d'email le 28 oct. 2015 entre Jacques André et moi même.

typographie dynamique, ou variable, en qualifiant le dessin des glyphes et le comportement de la fonte. Par cet emploi, je ne désignerai pas une typographie animée, mais une typographie au dessin non statique. Afin d'éviter toute confusion dans les termes, j'utiliserai toujours le terme de typographie animée pour parler d'une forme typographique en mouvement. Au cours des vingt dernières années, la typographie a continué son transfert technologique, passant de l'hégémonie du support statique imprimé à celle des supports dynamiques numériques, désormais incontournables. Sans affirmer, comme l'écrivit le studio IA en 2006 [w25], que 95 % du Web est composé de texte, nous conviendrons qu'une part essentielle de l'information visuelle présente sur le Web est typographique. Pourtant le dessin typographique reste figé, sans possibilité d'évolution de sa forme, mis à part une simple mise à l'échelle. L'histoire de la typographie s'étale sur un peu plus de cinq siècles. La typographie digitale est jeune d'une quarantaine d'années environ. Nous pourrions en déduire que le statisme de la typographie découle de sa jeunesse. Il n'en est rien, de nombreuses expérimentations de fontes dynamiques ont été menées depuis la naissance de la typographie digitale, mais aucune n'a été massivement adoptée par le public. On pourrait donc aisément en déduire une victoire naturelle du format statique. La typographie dynamique serait-elle une fausse bonne idée ?

Dans le livre *Du plomb à la lumière*, Alan Marshall raconte l'histoire de la photocomposition, et répond à l'idée d'une sélection naturelle dans la technique.

Quant à l'idée selon laquelle la technique « optimale » va toujours finir par s'imposer en vertu d'une sorte de processus de sélection naturelle, l'histoire des techniques ne manque pas de contre-exemples démontrant que ce n'est pas forcément la technique la plus performante qui connaît le plus grand succès sur le plan économique. [...] Force est de reconnaître que l'innovation n'est pas un processus linéaire gouverné par une logique implacable d'optimisation technique. Certes, la tendance est généralement vers une plus grande efficacité (notion qui doit être comprise dans le sens économique, mais aussi technique). Mais sa progression est souvent plutôt arborescente, son chemin étant jalonné de choix qui s'ouvrent aux ingénieurs et aux entrepreneurs au gré des opportunités techniques supposées optimales. [B18]

Il n'est pas question de dire que les formats de fontes statiques sont mauvais ou inférieurs à ce que pourrait être une typographie dynamique. Au contraire, il me semble que la typographie digitale est arrivée à un niveau permettant une expressivité et une qualité dans la production et l'utilisation de typographie jamais égalée. C'est justement cette qualité et la résolution de la plupart des problèmes propres au format typographique numérique qui me permet de penser qu'il est aujourd'hui temps d'avancer vers une autre typographie, plus fluide et dynamique.

B Dynamique du format et de la lettre

La typographie est « l'art d'agencer les textes avec des caractères, *types*, pour l'imprimerie ou pour tout autre support. » [1]. Un type, du latin *types* étant un « modèle, une figure originale » [1], à la différence de l'écriture, « Procédé qui permet de représenter un langage avec des symboles ou des lettres. ». [1] On comprend aisément l'importance du caractère « type ». Il est le modèle de base à partir duquel seront produits ses multiples. Pourtant il convient de ne pas opposer écriture et typographie. Après tout, la seconde est une sédimentation de la première, et on peut même parfois parler d'écriture typographique. Gerrit Noordzij [B10. 1] fait le lien entre ces deux disciplines sœurs en donnant sa propre définition de la typographie, « [...] l'écriture à l'aide de lettres préfabriquées » [B21].

Cette définition fait écho au principe même de la presse typographique de Gutenberg qui consistait à aligner des caractères en plomb pour ensuite imprimer des exemplaires d'un même texte. Née avec la modernité et Gutenberg en 1450, la typographie est animée par l'esprit de la Renaissance. [B22] La naissance du livre] Elle est un outil au service de la diffusion des savoirs et des connaissances de l'époque. Jusqu'alors, les textes étaient retranscrits manuellement par des moines copistes, un travail laborieux, long, coûteux et favorisant l'apparition d'erreurs entre les copies. Pour permettre la mécanisation de l'écriture, celle-ci a dû être rationalisée. Avec

[1] Définitions extraites de <fr.wiktionary.org>

Gutenberg naît le type, le modèle à partir duquel est produit l'ensemble des caractères mobiles. Une fois un texte imprimé, les caractères sont redistribués dans leurs casses pour être réemployés. D'une écriture nécessairement singulière, on passe à la reproduction d'un modèle. Depuis Gutenberg, cette distinction n'a jamais réellement évolué et s'est même renforcée au fil des évolutions techniques. Pourtant, aujourd'hui la typographie est digitale et le dessin des caractères n'est plus contraint par le plomb ou quelques autres matériaux.

La quasi-totalité des caractères imprimés passe aujourd'hui par l'informatique. On ne peut plus parler de typographie « assistée » par ordinateur : les fontes sont des objets informatiques à cent pour cent, complètement dématérialisées. [B8]

Énoncée en 1992 dans l'Introduction de son habilitation, cette affirmation de Jacques André [B10 10] reste vraie. J'ajouterai même qu'aujourd'hui la majorité des textes lus par l'œil humain n'est plus imprimée, mais affichée par un dispositif d'affichage dynamique. Qu'il s'agisse d'un écran LED, LCD, Plasma, CRT, d'encre numérique, ou de toute autre technologie, l'apparition des dispositifs d'affichages digitaux a révolutionné notre manière d'appréhender le texte et nous a permis d'interagir avec lui. Un écran LCD affiche en moyenne 60 images par secondes (60 Hz). Ce n'est donc plus une, mais des milliers d'instances d'affichages du même signe qui peuvent être opérées pour chaque glyphes.

Notre expérience des méthodes de composition traditionnelle nous a amené à nous attendre à ce que les formes des lettres composant une même typographie aient toujours la même apparence. Cette notion résulte d'un processus technique, et non l'inverse. Cependant, aucune raison technique ne pousse une lettre digitale à être la même à chaque impression. Il est possible de calculer différemment l'ensemble des points et des courbes à chaque impression en déplaçant légèrement les points qui définissent un caractère dans différentes positions aléatoires. [W24]

Le changement est la règle en informatique, la stabilité l'exception. [W4]

Comme l'affirment Jay David Bolter [BIO 4] et Letterror [BIO 7], il n'y a plus de raison technique à ce que la typographie reste statique dès lors qu'elle est le fruit de calculs numériques. Traditionnellement, les caractères typographiques se traduisent par des formes visuelles figées. A contrario, leur immatérialité devrait en faire des objets d'une extrême malléabilité. Il faut dire que la typographie a toujours été guidée par la quête de la perfection de la reproduction de ses types. Cette perfection de la reproduction des formes nous pouvons aujourd'hui la mettre en opposition — en l'observant a posteriori — à la forme du texte des impressions au plomb. En effet, si l'on ouvre un incunable du XVI^e siècle le texte s'anime, vibre. [IMG B3] De subtiles variations brisent l'homogénéité et la monotonie du gris typographique [LEX15]. Cette vibration, loin d'être un défaut, nous l'avons perdue au moment où la typographie est passée de trois dimen-

sions à deux. Pour l'expliquer, il faut comprendre le procédé typographique original créé par Gutenberg, se basant sur des modules reproductibles à l'infini. Afin de créer des caractères mobiles facilement reproductibles — les types — on crée des matrices. Ceux-ci sont formés à partir d'un poinçon [IMG C13] gravé préalablement par le graveur de poinçons qui vient frapper les matrices pour les marquer son emprunte. Les poinçons eux-mêmes sont parfois sculptés par la frappe d'un contrepoinçon, notamment dans le cas de contreformes [LEX16] se répétant d'un glyphe à un autre, comme la contreforme du « p » ou du « b ». Les matrices sont ensuite intégrées dans un moule dans lequel on déverse l'alliage qui, une fois refroidi, formera le plomb typographique [V5].

Ce principe devrait avoir pour effet qu'un « a » présent dans une casse sera toujours le même pour un corps de texte donné. Pourtant on ne trouvera pas — après impression — deux « a » parfaitement identiques. Au moment de l'impression, de nombreux paramètres viennent altérer l'homogénéité des types. La qualité du papier, l'usure des caractères, la répartition de l'encre sur les plombs et la force exercée par la presse sont autant de facteurs qui, cumulés, génèrent une fluctuation de la forme des glyphes à la surface du papier. Au fil du temps, les imprimeurs tentèrent d'atténuer ses variations. On les retrouve donc plus ou moins fortement tout au long de l'histoire de la typographie au plomb, suivant les évolutions techniques. L'état de la typographie telle que nous la connaissons aujourd'hui s'inscrit en quelque sorte dans une forme d'aboutissement technique de l'histoire de la reproduction des types. En effet, même s'il reste pos-

AVLI GELII NOCTIVM ATTICVM
CAPITVLA PRIMI LIBRI.

Vali proportione: quibusq; collectis
esse Pythagorā philosophū dixerit
proceritate: qua fuit Hercules: cum uita
Caput primum. Plutarchus.

Ab Herode Attico consulari uiro tempe-
riacatum & gloriosum adolescentē speci-
uerba Epicteti stoici: quibus festiuiter a

— Oui.

— Comment s'appelle-t-elle, Ordo ? »

Un petit silence : « La phonologie. »

Hilsh met la console sur VISUEL et t
NOLOGIE. Les informations défilent s
« Mais ça concerne le mécanisme de la
les sons. Je veux quelque chose sur les m

— Quels mots ? » répond Ordo à sa de

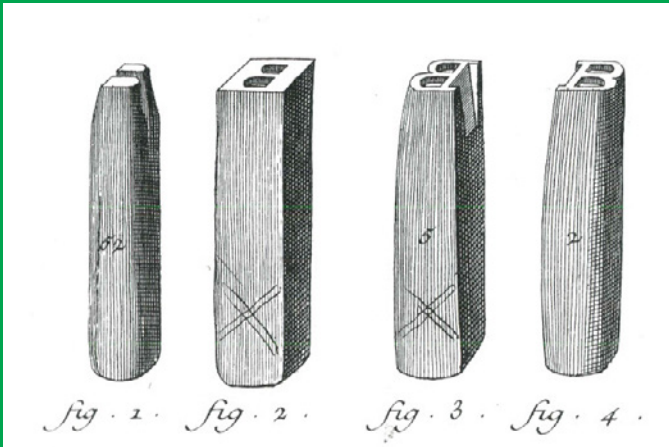
« La, une, elle. Des définitions de
Ordo », dit Hilsh après avoir remis la c
VOCAL.

Un petit silence de nouveau ; Hilsh se



FIG. 11-1. Simplicity of handling has established the Linotype slug as the modern unit of composition.

[IMG A17]



[IMG C12]

sible de trouver des micros variations dans la forme des glyphes imprimés par une imprimante laser à haute résolution, celles-ci seront invisibles à l'œil nu.

La technique peut influencer, par ses limitations ou ses fonctionnalités, la forme d'un dessin typographique. Ainsi l'évolution qui a mené à la mécanisation de la composition [IMG D1 - 13], de même que l'évolution technique des presses, de la qualité du papier et de la gravure de poinçons [IMG B1 - 6], ont influencé l'histoire typographique en permettant la gravure de caractères aux formes de plus en plus fines, une typographie au gris de plus en plus régulier, un contraste et une précision des formes imprimées de plus en plus importantes. Ces avancées techniques eurent une grande influence sur les formes typographiques, rendant ainsi possible la réalisation de dessins extrêmement contrastés comme celui de Didot. Le passage de la typographie d'un support physique à sa dématérialisation a eu le même effet, libérant les formes de leurs contraintes physiques, mais les contraignant parfois au travers d'une grille de pixels ou une technique d'impression. L'ouverture de la typographie a une variation de la forme des glyphes aura nécessairement une influence sur le design des caractères. En multipliant les variantes des dessins, en questionnant ce qui a été dessiné et ce qui a été généré, en permettant de nouvelles formes textuelles, les designers auront certainement l'occasion de se réappropriier et d'explorer de nouvelles formes.

L'utilisation d'une fonte au dessin variable, en plus de permettre aux glyphes de jouer un rôle dans la mise en espace

du texte, pourrait permettre de retrouver cette vibration. Dématérialisée, la typographie est « simulée », dessinée sur l'écran par un moteur de rendu [LEX22] qui lit les instructions contenues dans le fichier fonte. Si deux dimensions se trouvent simulées, ne pourrait-on pas rendre compte de la troisième ? Dès lors, pourquoi se limiter à trois dimensions ? Pourquoi ne pas en envisager quatre, cinq, dix, douze ?

Cette notion d'espaces ou de dimensions de variations, nous la retrouverons dans le second chapitre de ce texte à travers l'exemple des axes d'interpolation. Ainsi, de la même manière qu'un italique et un romain sont deux dessins pouvant cohabiter dans un texte, pourquoi ne pas penser l'utilisation d'un texte au gris dynamique comme un nouveau niveau hiérarchique ? Francesco Griffo, lorsqu'il dessina le premier italique pour les presses Aldines au XVE siècle n'avait pas envisagé l'utilisation de ceux-ci en combinaison avec le Romain. L'inclinaison du dessin des glyphes avait alors pour seul but de répondre à un problème pratique et économique : faire entrer davantage de texte dans l'espace de la page (le format « octavo ») afin d'économiser du papier, très cher à l'époque. En faisant varier le dessin des glyphes, on s'attend à pouvoir faire évoluer la taille de ceux-ci afin qu'ils s'adaptent à l'espace dans lequel on veut les contraindre. On s'attend également à pouvoir faire varier le gris typographique. Le réinvestissement de ce langage hérité de la typographie de la Renaissance viendrait alors se poser en rupture avec le gris homogène de la typographie statique et de l'impression à plat, créant de ce fait, une nouvelle forme plastique et hiérarchique du texte.

C Définir le dessin par le programme



Felzer Feliciano,
Cité du Vatican 1460



Damiano Molle, Parnis 1460



Felzer Feliciano,
Cité du Vatican 1460



Damiano Molle, Parnis 1460



Hartmann Schedel, Munich 1482



Luca Pacioli, Venetia 1509



Hartmann Schedel, Munich 1482



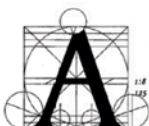
Luca Pacioli, Venetia 1509



Sigismondo Fanti, Venetia 1514



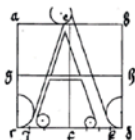
Francesco Tomiello, Milano 1517



Sigismondo Fanti, Venetia 1514



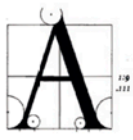
Francesco Tomiello, Milano 1517



Albrecht Dürer, Nuremberg 1525



Albrecht Dürer, Nuremberg 1525



Albrecht Dürer, Nuremberg 1525



Albrecht Dürer, Nuremberg 1525



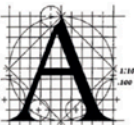
Giovan Battista Venturi, Tizolano 1526



Geoffrey Tory, Paris 1529



Giovan Battista Venturi, Tizolano 1526



Geoffrey Tory, Paris 1529

[IMG C4]

72 pt 48 24 18 14 12 10 8 6 5 4

aaaaaaaaaaaaaa

[IMG B15]

La forme finale que nous percevons actuellement de la typographie — dans le cadre d'un texte destiné à être lu — est une forme statique, qu'elle soit imprimée ou affichée. Cette forme statique oblige le designer à définir un dessin précis qui sera lu et vu par le lecteur. Nous venons de le voir, historiquement ce dessin était réalisé directement via la gravure de poinçons. [IMG D19] Assez rapidement, dans un souci de plus grande maîtrise, les lettres furent préalablement dessinées sur un support plat à grande échelle. Ce dessin de la lettre à plat amorça la distanciation entre les formes de l'écriture manuscrite — canon de la typographie — et leur contrepartie typographique. [IMG C1 - 15] Ces dernières tendirent progressivement à être rationalisées. Ainsi quelques décennies après Gutenberg, le modèle de la lettre a peu à peu été géométrisé et l'alphabet gothique utilisé dans les premiers livres fut supplanté par la lettre romaine, construction de l'esprit humaniste de la Renaissance [B22].

Faut-il donc tant de carrés pour former un O, qui est rond, et tant de ronds pour former d'autres lettres qui sont carrées ? [1]

Le passage de la gravure des types à leur dessin sur plan engendra une perte d'échelle optique. Or, la perception d'une forme graphique n'est pas là même selon l'échelle et la distance à laquelle notre œil se trouve. Pour cette raison, il faut adapter le dessin en fonction de sa taille

[1] Fournier Le jeune, *Manuel typographiques*, Paris, 1764

afin de permettre une perception idéale, compensant les effets d'aberrations optiques perçus par notre œil et notre cerveau. [w49] Assez rapidement dans l'histoire de la typographie, le pantographe fut utilisé pour permettre la conversion mécanique d'un dessin à grande échelle en un dessin de petite taille [w50].

Cet outil inventé en 1603 par Christoph Scheiner, un astronome allemand, fut très utilisé pour la production typographique à partir du XIX^e siècle. [ANN1] La dématérialisation progressive de la typographie a décuplé cette perte d'échelle en détachant le dessin des glyphes de toute forme tangible. Dans *Les contrepoinçons* [B15], Fred Smeijers [BIO14] témoigne de son expérience de la gravure de poinçon qu'il met en regard de la pratique du dessin de caractère digital. Selon lui, le gain de précision apporté par ce dernier nous éloignerait parfois de l'essentiel. En définissant le tracé de manière trop précise, à la recherche d'une courbe parfaite permise par la dématérialisation de la lettre, nous nous éloignerions du but de la typographie en nous attachant à des détails superficiels jusqu'alors invisibles de l'œil humain. En effet, le travail à l'échelle du dixième de point typographique [LEX19] était inconcevable techniquement à l'époque de la gravure de poinçons. À cette échelle, le designer aura alors davantage tendance à croire ce que lui disent les chiffres et son écran plutôt que son propre regard sur le texte à son échelle d'utilisation. En effet, c'est à l'échelle du mot et du texte — et non du glyphe — que se juge un dessin de caractère. Mais si l'existence numérique des caractères peut mener les designers à se perdre dans les méandres de détails absurdes, la malléabilité offerte par l'outil informatique doit per-

mettre d'y remédier. En effet, le dessin de caractères typographiques oblige le designer à penser les glyphes de manière globale. Ce travail le force à de nombreuses tâches répétitives et à de nombreux aller-retour entre les glyphes afin d'homogénéiser les noirs, d'équilibrer le rythme et les courbes de chacun d'entre eux. Or ce travail de longue haleine, réalisé optiquement glyphe par glyphe, peut désormais être en partie automatisé par le recours au code informatique. La description digitale de la forme des glyphes a pour effet de pouvoir facilement modifier le positionnement et la forme d'un grand nombre de glyphes à la fois. Les accents des glyphes accentués par exemple ne sont plus réalisés un par un à la main. Ils découlent d'un module répété et positionné de manière fluide sur l'ensemble des glyphes concernés. Ainsi qu'il s'agisse des accents, de la largeur d'un fût [LEX17], de la hauteur des montantes [LEX18] ou encore de l'inclinaison d'un tracé, le designer peut faire découler le dessin de ses glyphes d'une logique propre à l'ensemble de la typographie, voir de la famille typographique sur laquelle il travaille. Cette fluidité offerte par la nature numérique de la typographie permet une plus grande mobilité dans le travail du designer. C'est cette mobilité qui lui permettra de se concentrer davantage sur la globalité de la typographie et de réaliser de nombreux essais à l'échelle du texte.

Si vous concevez une seule fonte, c'est une île. Si vous en concevez plus d'une, vous êtes en train de concevoir leurs relations, la recette. [w22] [c5]

Mais peut-on encore parler de dessin ou de la forme d'un caractère dans le cadre de la création d'une fonte dynamique ? Dans *Designing Programmes*, Karl Gerstner [B21] parle de la conception de la typographie comme d'un programme visuel, découlant de décisions et de paramètres. Il n'y est pourtant pas question de typographie dynamique.

La famille de caractère *Univers* [IMG E1] de Adrian Frutiger [BIO 15] montre qu'il n'a pas fallu attendre l'arrivée de l'informatique pour penser la typographie comme un programme visuel. Avec l'*Univers*, Adrian Frutiger envisage la création de 21 caractères composant une même famille autour d'un même programme. Il utilise même un système alpha numérique afin de le nommer et les classer par ordre de largeur et de graisse. Dans le cadre d'une typographie dynamique, le programme visuel dépasse le système visuel propre à la famille typographique. Le designer ne dessine plus un tracé fini, mais un système de formes pouvant varier. Il devient alors possible d'utiliser un caractère typographique à la couleur et au rythme changeants.

Cet aspect a été compris depuis assez longtemps par les designers typographiques ainsi qu'en témoigne cette citation de Jan Middendorp [BIO 8] extraite d'un article traitant du design paramétrique :

Un design commence habituellement avec une série de croquis, une vague visualisation des possibilités qui laisse le processus de sélection — le choix entre les bonnes et médiocres solutions — se faire. Tant que vous continuez de dessiner sur un coin de table, ces notes

peuvent signifier ce que voulez. Mais si vous commencez directement avec une souris dans la main, ce qui est plutôt commun de nos jours, vous êtes immédiatement confronté à des dixièmes de millimètres et des lignes parfaitement droites. Le design prend déjà une forme définitive alors que c'est le moment d'être flexible. [...] Il existe une alternative : le design assisté par la programmation. Dans ce cas, les différentes possibilités ne sont ni croquées à la main ni dessinées directement sur l'ordinateur. Le designer indique les ingrédients et les marges, et demande à l'ordinateur d'inventer une série de variantes en prenant en compte certaines règles. En d'autres termes, il lui demande de faire varier certains paramètres aléatoirement. L'ordinateur ne conçoit pas lui même, il ne peut donc pas être accusé d'interférer avec le processus créatif. Il fait ce qu'un ordinateur fait le mieux : appliquer très rapidement les conséquences de possibles décisions. [W40]

De nombreux outils sont aujourd'hui développés afin de permettre l'automatisation de certaines tâches dans le dessin de caractère, à l'image du programme développé par Andy Climer — dans le cadre de la création du caractère *Obsidian* [IMG D35] — qui permet de générer le dessin d'un effet d'ombre inspiré des fontes décoratives en usage sur les cartes postales au XIX^e siècle. La plus grande partie de ces programmes informatiques n'est pas distribuée, car réalisée à des fins précises. Ces programmes informatiques, du plus simple au plus complexe, sont destinés à la conception de fontes et sont donc utilisables à l'échelle du glyphe. Ils ne s'adressent donc pas aux designers graphiques. Ce point sera abordé plus tard.

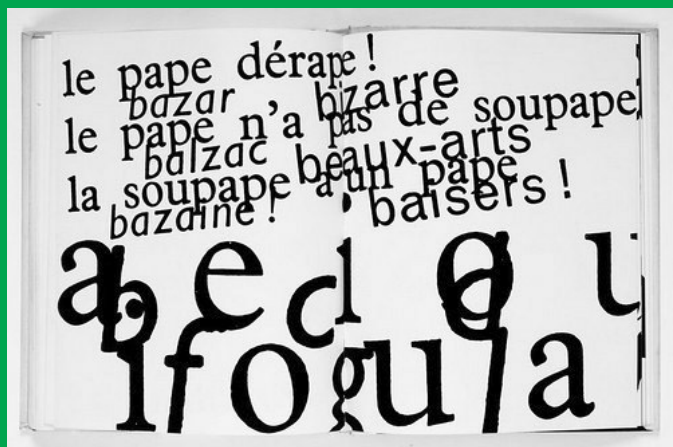
Si le dessin de l'ensemble des glyphes d'une fonte peut facilement être modifié, alors il devient envisageable de voir la forme de ses glyphes varier au cours d'un texte, ou dynamiquement dans le cadre d'une page Web, une fenêtre ou toute autre surface d'inscription à l'écran. Une telle typographie pourrait-elle reproduire une écriture manuscrite ? De nombreux projets ont été menés avec cette ambition.

Le *Zapfino* [1] d'Hermann Zapf [BIO 17] fut l'une des premières fontes atteignant partiellement ce but en proposant des caractères typographiques s'imbriquant parfaitement les uns dans les autres pour former une ligne continue simulant le tracé de l'écriture au fil de la plume. Cependant, ce qui caractérise la typographie est sa perfection dans la reproduction des types. Or, ce sont justement les imperfections qui font, à travers leurs rythmes et variations, l'essence de l'écriture humaine.

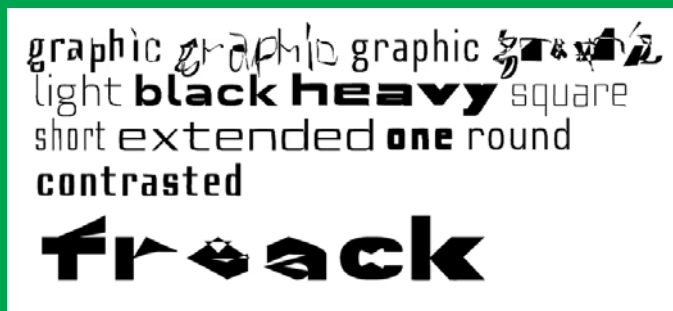
Avec une typographie immatérielle on peut imaginer le recours à l'aléatoire et aux algorithmes pour simuler des défauts. C'est en partie le projet du *Beowolf* [IMG E7] de Letterror [BIO 7], un projet que je développerai plus tard. Mais l'aléatoire n'est-il pas trop chaotique pour reproduire ou simuler les variations subtiles de l'écriture manuscrite ? Au cours de la jeune histoire de la typographie digitale d'autres projets ont proposé des alternatives algorithmiques tentant de simuler l'écriture.

[1] Le *Zapfino* est fonte PST1 publiée en 1998. Elle s'inspire d'une calligraphie réalisée par H. Zapf en 1944.

D Texte en mouvement



[IMG E15]



[IMG E14]

Le texte est de plus en plus mobile. Au-delà de la fluidité des espaces dans lesquels il s'inscrit, le sens même du texte peut se trouver continuellement altéré. Sa forme, statique porte des messages changeants sans jamais en être affectée ou sans en notifier son lecteur. Bien que l'on associe à l'écriture, et plus particulièrement à la typographie, les termes de stabilité voir d'immuabilité, le texte semble aller vers une forme de plus en plus instable. Sans cesse modifié, en constante évolution, le statut de l'écrit en est transformé. À ce titre, l'écrit se rapproche de plus en plus du langage parlé. Dans *La forme solide du langage*, Robert Bringhurst [BIO 12] tente de cerner la matérialité du langage :

La parole sort de nos bouches, de nos mains, de nos yeux comme une forme liquide et s'évapore aussitôt

[B4]

La typographie telle que nous la connaissons aujourd'hui — statique — est l'une des formes les plus solides du langage. Elle a pourtant perdu aujourd'hui une grande partie de sa stabilité et on peut constater une forme de « liquéfaction » de sa forme. En effet, avec l'écriture calligraphique, le tracé de l'outil vient marquer le geste, l'intention. Il s'inscrit dans un espace temporel et physique. Originellement, la typographie est un art qui ne se pratique pas au moment de l'écriture, mais une fois le texte

clairement établi. Aujourd'hui, la plupart des textes sont composés directement sur ordinateur [IMG A22] sans passer par un stade manuscrit [IMG F5]. Pourtant les formats de la typographie digitale ne laissent guère de part à l'expressivité de l'écrivain ou à une possible interaction avec son lecteur. La question d'une typographie permettant une retranscription tenant compte de l'expressivité et de la richesse du langage non textuel de son auteur est une question qui a déjà été posée à de nombreuses reprises. Que ce soit à travers les poèmes métriques, tels *Un coup de dé n'abolira jamais le hasard* [IMG E13] de Stéphane Mallarmé, les calligrammes de Guillaume Apollinaire [IMG E14], la *Typoésie* de Jérôme Peignot ou plus récemment des expérimentations typographiques telles que le projet de Benjamin Gomez [IMG E15] proposant une interaction entre l'auteur composant son texte et le formatage du texte voir la forme graphique des glyphes. Chacune de ces expérimentations joue avec le texte et les formes de la typographie d'une différente manière. Qu'il s'agisse de sa mise en forme dans l'espace de la page, ou de la forme même des caractères, toutes entretiennent un rapport avec la forme, le rythme de la typographie, et donc du langage, mais aussi avec l'espace, celui de la page, du livre ou de l'écran.

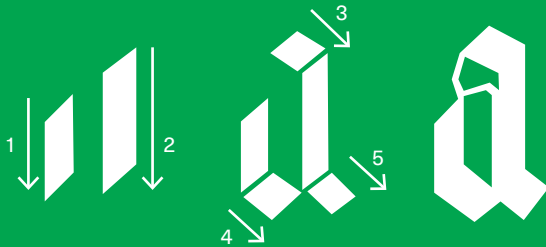
Ce mémoire ne s'intéresse pas à la question d'une retranscription de la richesse du langage à travers la typographie. Cependant, il est intéressant de constater à travers ces exemples comment le potentiel plastique de la lettre a pu être exploité à travers la variation de son dessin pour véhiculer un message sur sa nature. Cependant, l'interaction entre la lettre et le sens qu'elle véhicule n'a pas à être lit-

térale. Il n'est pas question ici de vouloir transmettre une émotion ou une idée via la visualité du texte, mais simplement de permettre l'énonciation d'un changement dans la nature de l'information, ou la valorisation d'un texte par une intervention minimale. Le rythme transmis par les glyphes passe nécessairement par leur rapport avec l'espace dans lequel ils s'inscrivent. La régularité ou l'irrégularité d'un caractère influence la lecture : le gris de la page, la forme des caractères, le rythme des blancs et des noirs, et enfin l'harmonie entre tous ces paramètres. À travers l'histoire du *Méridien*, Adrian Frutiger [BIO 15], parle de son rapport à la régularité.

Williamsann [un de ces anciens professeurs] était très enthousiasmé par le « Jenson ». D'emblée, son rythme régulier m'a fasciné. C'est une question de force de trait et de proportions. Pour un nouveau caractère, le plus dur est de trouver le rapport de force entre les valeurs blanches et les noires. Les écritures gothiques sont un modèle d'équilibre [comme la Textura [IMG B1]. Même les rondeurs débouchent sur des traits verticaux — ce qui gêne considérablement la lisibilité — et forment un maillage étonnamment régulier, avec des espaces vides homogènes. Mais également dans les Romains de la Renaissance [IMG B3], on note un effort pour harmoniser les blancs des contrepoinçons et ceux des espaces entre les lettres. [...] je dois dire ici que les lignes les plus harmonieuses ne sont pas forcément les plus lisibles. Une bonne lisibilité nécessite des formes de lettres variées, avec jambage, traits droits ou en diagonales, et courbes. [B12]

Si l'on en croit Adrian Frutiger, un rythme trop régulier ou une harmonie trop importante nuiront tous deux à la lisibilité et au confort de la lecture, de même qu'un rythme trop hétérogène et une absence d'harmonie dans la forme des glyphes. Ce constat pose de nombreuses questions, tant du point de vue de la relation du texte avec son espace d'expression qu'à celui du dessin des glyphes eux-mêmes. Mais n'oublions pas qu'Adrian Frutiger parle ici du dessin appliqué à la typographie au plomb. Il demeure intéressant de se demander quel effet pourrait avoir une typographie où le gris varierait au fil du texte. De même, quels effets auraient des rapports de blancs et de noirs inégaux au fil de la lecture ? Cependant la redéfinition des glyphes induit-elle de grands changements à l'échelle du texte ? Comment composer une ligne dont les glyphes peuvent changer de taille ? Comment décider où arrêter la ligne et gérer la forme d'un paragraphe aux dimensions variables ? Si un espace en perpétuelle évolution appelle à une redéfinition de la forme de la lettre, une lettre à la forme liquide implique, elle, une nouvelle relation avec l'espace qui l'entoure. Alors, comment négocier la place d'un texte variable à la forme dynamique dans une surface instable ? [IMG F1 - 9]

| | |
|---|------------------------|
| 2 | Espace formaté |
| A | Le texte dans l'espace |



L'inscription d'un texte dans l'espace d'un format — livre, rouleau, tablette, etc. — a une influence sur ce dernier. L'observation de formes graphiques telles que le traitement typographique des titres dans les journaux du début du xx^e siècle ou les lettrages des devantures de magasins rend parfaitement compte de l'enjeu. L'inscription d'un texte dans un espace donné soulève des questions visuelles complexes qui appellent à des résolutions différentes selon de nombreux paramètres.

Oublions un instant la typographie. Le calligraphe, lorsqu'il exécute une ligne de texte, prend en compte l'espace dans lequel il devra l'inscrire. Il doit alors manipuler plusieurs paramètres simultanément. D'une part, à travers la maîtrise de l'outil calligraphique il doit reproduire la chorégraphie donnant vie à chacune des lettres : le *ductus* [IMG c2]. Celui-ci comprend l'ordre et la composition des traits formant la lettre. Chaque variation dans le ductus aura pour conséquence une modification du dessin final de la lettre correspondante. Il doit également prendre en compte l'espace dans lequel inscrire chacune des lettres et anticiper la place que devra occuper chacune des lettres suivantes. En découle une suite de micro variations dans les tracés. Ainsi, le calligraphe adapte le ductus en prolongeant ou en raccourcissant les lignes, sans jamais rompre l'harmonie, le gris et le rythme de sa page. Ce sont ces légères modifications du dessin qui permettent au texte

de s'inscrire de manière naturelle dans la longueur qui lui est donnée. Bien entendu, cette opération, pour qu'elle soit menée à bien, demande une grande maîtrise calligraphique. De nombreux paramètres sont à prendre en compte, et l'espace noir et blanc propre à chacune de lettres les unes par rapport aux autres doit être connu à la perfection.

La description des caractères typographiques telle qu'elle est opérée aujourd'hui permet l'accès à toutes ces informations. En effet, comme je l'évoquais plus tôt, toutes les données relatives aux caractères sont contenues dans une base de données, au cœur du fichier fonte. [B13] Les valeurs métriques concernant l'espace blanc entre les lettres, leur largeur, la surface dessinée par le noir et l'interaction entre chacune des lettres permettent — une fois combinée avec la longueur de la ligne — de calculer de combien doit s'allonger ou se rétrécir la lettre pour remplir parfaitement la largeur de colonne. Actuellement, ces informations sont utilisées afin de calculer l'augmentation ou la réduction du blanc entre les signes ou entre les mots afin d'harmoniser les lignes de textes. Cependant, la modification du blanc du texte n'est pas sans risque. Utilisée de manière abusive, elle risque de créer des brisures dans le rythme du texte ou de créer des rigoles traversant sa hauteur, perturbant ainsi la lecture.

Pour réaliser cette tâche, des algorithmes sont utilisés. L'algorithme le plus connu et le plus répandu pour la gestion de paragraphe en typographie latine est le *Hz-program* (Herman Zapf program) de Peter Karrow [BIO 18, P.171] et Herman Zapf [BIO 17, P.171], sur lequel se base

notamment *Adobe inDesign*. Il permet une gestion automatisée de l'interlettrage, de l'intermot, le calcul automatique d'une justification ou d'un drapeau [B3]. À ce titre, il me semble intéressant à ce titre de citer le projet *Font-to-Width* de Nick Sherman [BIO 20] en collaboration avec Chris Lewis. Ce site Web [w26], réalisé en 2014 propose d'adapter automatiquement le choix de la fonte à l'aide d'un script afin que le texte occupe le maximum, voir la totalité de l'espace offert par la largeur de la colonne. S'appuyant sur les standards du Web en cours, Nick Sherman utilise une « super famille » [LEX11] au format *Opentype* : le *Panorama* de Jean-Baptiste Levée. Le nombre de variations en largeur présentes dans le *Panorama* étant extrêmement élevé, le redimensionnement de la page donne l'impression d'une interpolation du caractère dans le navigateur. Ce n'est cependant pas le cas. La force visuelle d'un tel projet est évidente. En automatisant par la programmation le choix du fichier fonte adapté au contexte, on permet la réalisation de formes jusqu'alors invisibles à l'écran. [1] Cependant, de nombreux alphabets non latins sont basés sur une construction plus calligraphique et ne permettent pas, dans leur construction, la justification par le blanc. Ces écritures nécessitent donc une justification dite sémitique par le noir [IMG F9].

[1] Cette approche de la gestion de la mise en page par le code est expérimentée par le projet *Universal Typography* où Tim Brown explore des façons d'adapter automatiquement les réglages typographiques (sans changer leur forme) à leur contexte. <universaltypography.com>
On peut également citer *Hypher*, un portage de l'algorithme de gestion de paragraphe de Peter Karrow sur le Web. <github.com/bramstein/hypher>

Permettre la justification par le noir, oblige le recours à une modification du dessin ou à la combinaison de formes s'emboitant les unes dans les autres ou se substituant les unes aux autres. Le formatage du texte par le noir est le formatage naturel du texte issu de l'écriture, la calligraphie, la peinture, la gravure ou toute autre technique mettant en jeu le corps. Qu'il s'agisse du lettrage d'une devanture, des gloses dans un codex, d'une lettre manuscrite ou des ligatures sur une inscription lapidaire, la lettre et le blanc qui l'entoure sont déformés pour s'inscrire dans l'espace qui leur est donné [F1]. Actuellement, ceci est permis par la technologie *Opentype* combinée à un moteur de justification de lignes et des fonctions propres à la typographie arabe. Celles-ci ne sont donc utilisables que dans le contexte d'un logiciel les prenant en compte. La suite *Adobe* classique ne prend pas en charge ces fonctionnalités, il faut recourir à ses versions *Middle East* pour y avoir accès. Le moteur de justification de ligne est un programme informatique basé sur un ou plusieurs algorithmes ayant pour but de décider de l'endroit où devra se terminer la ligne et, dans le cas d'une justification, combien de blanc ou de noir devra être ajouté à la ligne pour une optique optimale. Il est d'ailleurs intéressant de mettre en perspective l'effet qu'a pu, et continu d'avoir, le fait de déléguer la prise de décision de mettre fin à une ligne. Longtemps, cette décision fut la responsabilité du composeur (celui qui aligne les caractères de plombs). Elle était même l'une des compétences essentielles requises pour pratiquer ce métier. L'apparition de la mécanisation de la composition typographique — via les machines *Monotype* et *Linotype* [IMG A7 - 19] — posa le problème de l'automatisation de cette prise de décision.

Si la *Linotype* laisse le choix au composeur (sur clavier) de mettre fin à la ligne quand bon lui semble, la *Monotype* utilise un système de découpage des caractères en 18 unités. Ce découpage se traduit par une contrainte du dessin des caractères, devant à la fois se limiter et remplir ces espaces normés. En découplant les caractères en tranches de formats connus il devient alors simple — en additionnant la suite de valeurs de chaque glyphe utilisé, lui-même composé d'un certain nombre de tranches définies — de connaître la longueur du bloc en cours, et donc le moment auquel terminer la ligne et de chiffrer le nombre d'espaces blancs permettant de justifier le texte (la justification par le noir n'existait pas sur les machines de composition au plomb). La perspective offerte par l'histoire de l'automatisation du formatage du texte est intéressante lorsqu'elle est mise en relation avec les algorithmes actuels. L'automatisation du formatage a d'abord été réalisée dans le but d'augmenter le rendement en facilitant le travail des composeurs. Or cette automatisation est à l'origine de l'idée d'une typographie pouvant voir son dessin, et donc son format, se modifier au court de son utilisation. Il est d'ailleurs important de noter que le passage d'une typographie fixe à une typographie dynamique a pour conséquence de devoir repenser, voir d'adapter, les algorithmes de mise en forme du texte. Ce travail amènera probablement à changer notre manière de penser l'outillage nécessaire à son formatage. La contrainte de l'inscription du dessin des glyphes dans une grille normée pour le système *Monotype* est également intéressante par les conséquences qu'elle engendra sur le dessin des glyphes. En effet, conjugué à la nécessité de fondre différentes tailles de corps, le dessin des plombs

devait être adapté. On trouvait ainsi un n d'une largeur plus étroite en 12 points qu'en 7 points. N'y voyez pas là une correction optique, mais l'obligation de faire entrer un même dessin dans un nombre de modules aux dimensions indépendantes de l'échelle du caractère [IMG D4].

Ce principe de découpage des glyphes en modules de taille fixe a par la suite été repris dans la photocomposition. Il ne s'agissait plus là de 18 mais de 36 compartiments [IMG D5], libérant ainsi le dessin des caractères d'une grande partie de la contrainte. De plus, la projection de l'image des caractères à travers de « mini pochoirs » rendait inutile et complexe la réalisation de dessins spécifiques à chaque corps optique d'un caractère. Pour permettre un agrandissement d'un glyphe, il suffisait de changer l'optique de zoom et de mise au point par laquelle passait la lumière. Mais cet agrandissement sans mise à l'échelle optique du corps des caractères, similaire à celui opéré par le pantographe, conduit à la perte d'une partie de la richesse optique des caractères en plomb. La période de la photocomposition n'a pas duré longtemps ; elle n'a été qu'une transition dans l'histoire de la typographie. Pourtant ce moment a été très important, car il formalisa, avec l'utilisation de l'agrandissement photographique, les premières formes d'interaction immatérielles avec la typographie. Par la suite, l'informatique a permis de systématiser ce qu'avait commencé à faire la photocomposition. Ainsi, les technologies numériques permettent aujourd'hui de déformer le dessin des glyphes en largeur, en hauteur ou même de les incliner. Mais si les logiciels de publication assistée sur ordinateur (PAO), ainsi que les navigateurs Web, permettent cette déformation,

cette utilisation dans un texte est vivement critiquée, voire expressément interdite dans le milieu du design graphique et de la typographie, mis à part un usage radical ou expérimental assumé. La raison de ce regard sévère sur l'usage de ces déformations s'explique par les aberrations visuelles qu'elles peuvent causer. En transformant de manière linéaire le dessin des glyphes, le minutieux travail de réglage optique réalisé par le designer typographique est détruit. Cette fonctionnalité, introduite pour la première fois par la photocomposeuse de troisièmes générations *Fototronic CRT* voient le dessin des glyphes traités torturés et dénaturés par l'intervention électronique [B19]. Il est d'ailleurs intéressant de noter que cette fonctionnalité fut dans un premier temps utilisée par l'algorithme Hz — que j'ai évoqué plus tôt — afin de faire varier la largeur des caractères et équilibrer les lignes de texte. Pour comprendre la raison de cette gêne, prenons l'exemple d'un « a » que nous désirerions élargir. Ce « a » verrait la graisse de son fut déformée en largeur, il en résulterait en une graisse trop importante. Dérangeant si on la compare aux lignes horizontales de la lettre qui, étirées en longueur paraissent désormais trop fines... Il est malgré tout possible pour le designer typographique d'anticiper cette déformation en pensant le dessin des formes de manière à limiter les aberrations. C'est le cas du *Swift* et du *Gulliver* [IMG C31] de Gerard Unger [BIO 13]. Conçu pour l'impression de journaux, leur dessin évite au maximum le recours aux courbes aux endroits supportant mal les déformations afin d'éviter l'apparition d'aberrations dans le dessin et facilitant ainsi le travail de justification du texte des maquettistes en leur permettant, dans une moindre mesure, leur utilisation. Cependant, un

tel travail contraint énormément les formes et n'est pas applicable à tous les dessins de caractères. Il faut donc trouver une solution plus adaptée, permettant d'adapter la forme du texte à un espace, sans pour autant dénaturer son dessin.

Dans *Le trait, une théorie de l'écriture* [B20], Gerrit Noorzij [B10 1] propose d'envisager la description de la typographie non par les contours définissant le dessin des glyphes, mais par leur tracé central, leur ductus. Cette approche précédant l'apparition des éditeurs PAO et de nos formats de fontes actuels se distingue d'une approche historique du dessin de caractère. En effet, la description par les contours précède largement l'invention de la typographie digitale. Dès l'origine de la discipline, la gravure des poinçons consistait en la délimitation des contours des surfaces formant la lettre. De ce fait, la création d'un caractère passa rapidement par un dessin définissant ses contours [IMG C1]. Ainsi, la nature mécanique de la typographie a vu la malléabilité de la lettre s'amoinrir et s'éloigner du dessin calligraphique. La définition des contours et leur gravure ont mené à une redéfinition géométrique et mathématique [IMG C5], voir parfois ésotérique [IMG C3] de la lettre. Il faut dire que le mouvement humaniste influença la typographie dans sa manière d'envisager la définition des formes et des lettres. Une fois définis, les contours sont invariables et reproduits lors de la gravure des poinçons. Actuellement, la plupart des programmes de création de fontes proposent un dessin des glyphes par leurs contours. Cette approche semble en effet logique, quoi de plus naturel pour définir une forme ? En affirmant, a contrario, que la forme écrite de

chacune des lettres découle d'un tracé accompagné d'un certain nombre de paramètres tel que la forme de l'outil et son angle d'inclinaison, Gerrit Noordzij fait la différence entre deux types d'écritures.

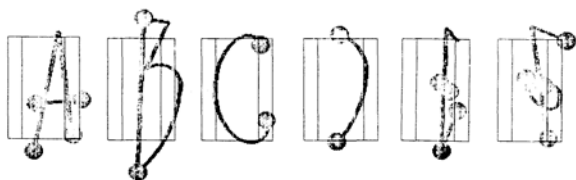
L'écriture manuelle et l'écriture dessinée. L'écriture manuelle est l'écriture du trait simple. L'écriture dessinée est une écriture de formes dessinées. [...] L'écriture dessinée est indépendante de l'outil, mais cette liberté n'est possible qu'aux dépens du caractère : dans une écriture composée d'une superposition de traits, les formes des traits simples se perdent, tout comme les empreintes d'un pied se fondent dans la multitude des autres traces de pas. [...] Il n'y a rien de ça de la forme d'un trait simple. [...] Les contours sont les limites des formes. [B20]

Gerrit Noordzij montre qu'un dessin décrit par ses contours ne peut évoluer, contraint par les contours qui le ferme, le limite. Pour être modifiable, ouverte, la description de la lettre devrait donc passer par son tracé central. La proposition d'une lettre qui naitrait virtuellement des changements de paramètres de son tracé, telle que son inclinaison, la forme de son outil, la pression ou la rotation exercée sur ce dernier, permettrait en effet d'envisager une écriture modulable dans son espace d'écriture. Cette prise de position en faveur d'une description de la typographie par le trait, va dans le sens d'une évolution de la typographie vers plus d'adaptabilité. Elle a depuis été reprise dans de nombreuses expérimentations typographiques telles que les projets *Calliculator*, *Constructor* ou la plus connue et première d'entre elles, *Metafont* [IMG D25, P.314]. Pourtant,

20 ans après la parution de *The Stroke* [B20] c'est l'approche par le contour qui est massivement utilisée à travers les courbes de Béziérs. Il existe bien entendu de nombreux contre-exemples. L'italique dessinée par Francesco Griffo en est un. De même, les lettres de titrages en bois [IMG F7] utilisés massivement à partir du XIX^e siècle s'inscrivent dans une volonté d'optimisation de la relation entre le texte et son espace d'expression. Étirés en largeur ou en longueur, réalisés dans l'objectif de remplir la page ou l'affiche au maximum, ils occupaient une fonction signalétique s'approchant davantage du lettrage que du texte. La liberté de créer un caractère propre à un espace donné est donc possible, les exemples ne manquent pas, cette pratique est cependant réservée aux seuls designers typographiques. Donner la maîtrise de l'espace qu'occupe un glyphe à son utilisateur lui permettrait d'avoir un meilleur contrôle sur les formes typographiques qu'il utilise, d'enrichir son vocabulaire formel, et par extension la mise en forme de son texte. La typographie dynamique a beaucoup à apporter au formatage du texte, en faisant de la forme des glyphes une partie active, et non plus passive, car statique de celui-ci. Afin de mieux envisager la forme que pourrait prendre une typographie dynamique et ses effets sur le texte, il est important de se pencher sur les mécanismes internes propres à la typographie elle-même. Cela passe par une étude des formats de quelques approches de la typographie.

B Le format typographique

METAFONT output 1985.02.28.1504



(I refuse to show you the first form of the letter 'G'.) But by 4 p.m. I was ready to make the first trial setting of text:

THIS 'PUNK' ALPHABET, INSPIRED BY MARJAN'S LECTURE LAST NIGHT, WAS DESIGNED BY METAFONT'S RANDOM NUMBER GENERATOR.

THE QUICK BROWN FOX JUMPED OVER THE LAZY HAMBURGERS.

L'histoire de la typographie numérique a vu naître de nombreuses tentatives de description et d'utilisation de la typographie aujourd'hui abandonnées ou peu utilisées. L'une des plus connues d'entre elles est *Metafont*, œuvre d'un mathématicien américain dans les années 1980, Donald Knutt. Précédant l'invention du langage *PostScript* [ANN7], langage informatique de description de la page dominant aujourd'hui l'ensemble de nos formats typographiques, il se proposait de définir une typographie par son tracé central (le ductus) couplé à un ensemble de paramètres. Une des premières expérimentations de Donald Knutt fut la fonte *Punk* [B21] [IMG E5]. Créée en 1980 et publiée en 1988, celle-ci se voulait inspirée des graffitis *punks* des années 1970. Dans cette première tentative, D. Knutt illustre le concept qu'il essayait alors de mettre en place. Composé de lignes reliant des points qui marquent les points d'ancrage, ce caractère imite le tracé d'une écriture linéale, tracé à main levée. La description du tracé n'est pas réalisée par les contours de la lettre, mais par son squelette. Ce n'est bien entendu pas la forme — pour le moins rudimentaire — de la fonte qui nous intéresse ici, mais la démonstration des possibilités offertes par l'outil informatique dans l'affichage de fontes au tracé dynamique.

Lors du premier appel de chaque glyphe, ses points d'ancrage varient et modifient ainsi la forme. La fonte est donc différente à chacune de ses utilisations.

Ce projet préliminaire annonce *Metafont*, un projet plus large qui n'avait pas pour idée de créer une typographie au dessin constamment variable, mais une typographie au dessin adaptable dessiné à partir d'un tracé central : la calligraphie digitale qu'avait rêvé Gerrit Noordzik. Créé à l'origine par D. Knutt pour satisfaire son propre usage de la typographie digitale, qu'il jugeait alors d'une qualité insuffisante, *Metafont* avait pour objectif d'égaliser le rendu de la typographie au plomb pour l'impression de ses ouvrages de recherche. Une plume virtuelle dessine en suivant le ductus numérique, déposant du noir sur la grille pixel comme s'il s'agissait d'encre numérique. Ce procédé donne un résultat sous la forme d'un fichier image bitmap et non une description vectorielle de la forme. Si ce format s'avère être aujourd'hui extrêmement lourd et peu pratique, il était logique à l'époque, les imprimantes ayant chacune leur moteur de rendu et leur langage propre. Les courbes de Béziérs ne deviendront la norme pour le dessin typographique et son impression qu'avec l'apparition du format *PostScript*. Plus question de correction optique ou de mise à l'échelle des caractères, ceux-ci voient leur dessin généré sur mesure, pour leur contexte d'utilisation. Ces dessins dynamique permettent notamment la justification par le noir.

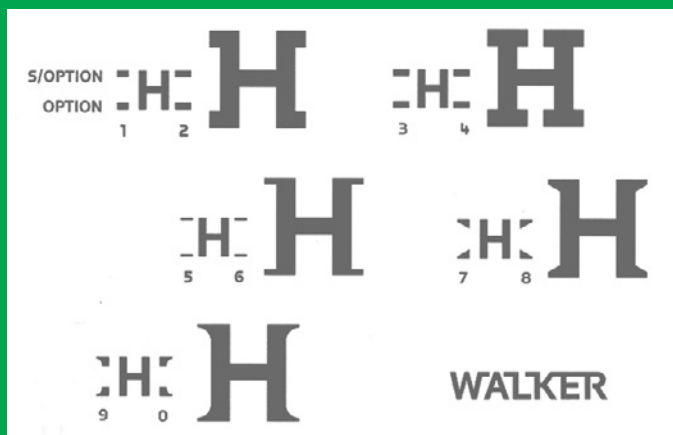
La première expérimentation de D. Knutt (*Punk*) fut reprise par Jacques André en collaboration avec Victor Ostromoukhov pour la traduire en *PostScript*. En effet, si le dessin du *Punk* était généré aléatoirement lors de l'installation de la fonte, il n'était ensuite plus modifié, rendant chaque glyphes identiques les uns aux autres

par la suite. Le but du portage sur *PostScript* était de prolonger le principe pensé par Knutt en régénérant le tracé de chaque glyphe à chaque instance d'utilisation. Pour ce faire, le cache [LEX20] dans lequel *PostScript* enregistre le dessin de la lettre après sa première apparition afin de ne pas avoir à la recalculer à chaque apparition fut contourné — rendant par là même la vitesse de calcul extrêmement lente. On obtient alors un texte où les glyphes sont tous différents les uns des autres. La même année, Letterror présenta dans la revue *Emigre* son projet *Beowolf* [IMG E7] utilisant lui aussi le langage *PostScript* via le format de fonte *PostScript Type 3* [ANN9]. Cependant, ces derniers préférèrent l'utilisation d'un dessin par le contour au dessin par le tracé central. Le contour y est défini par un ensemble de points aux coordonnées variées dans un périmètre défini par la fonction *freakto*. Cette fonction, créée pour l'occasion, fut nommée en référence à l'instance *PostScript* *lineto* permettant de relier deux points par une courbe de Bézières. Ces deux utilisations du langage *PostScript* faisant varier la position des points ont une limite : leur lenteur d'exécution. En interdisant l'utilisation de la mémoire cache la durée du calcul de l'affichage des glyphes peut-être multiplié jusqu'à mille, rendant ce système pratiquement inutilisable. Ces deux prototypes sont néanmoins riches d'enseignements dans leur approche conceptuelle du dessin des glyphes et du rendu visuel d'une fonte dynamique. *Metafont* s'inscrit dans un projet plus large, à l'échelle de la page. Associé au moteur de traitement de la page *Tex* il permet la variation du dessin des glyphes selon l'espace et le corps dans lesquels ils sont utilisés. Les premières expérimentations dynamiques de Donald Knutt utilisaient un mode de

tracés minimal. Celui-ci a rapidement évolué pour permettre l'utilisation de plusieurs « pinceaux virtuels » ainsi que le remplissage des tracés, rendant ainsi la plupart des formes envisageables. Cependant, la manipulation de tous ces outils, sans jamais avoir recours à une interface graphique, reste assez contre-intuitive et limite l'adoption de ce format dans le cadre d'une création de nouvelles fontes originales. Ainsi, bien que très apprécié pour sa rigueur dans la gestion des formes typographiques complexes, *Metafont* est peu utilisé en dehors du monde scientifique et universitaire. Pour pallier ces défauts, des tentatives d'évolutions et de simplification du concept ont été récemment menées. Ainsi, le récent projet *Metapalator*, basé sur les travaux de *Metafont* essaye de marier le meilleur de deux mondes : dessin par le tracé central et description par les contours.

L'idée de ce type de projet n'est pas de créer un environnement ou un format permettant d'utiliser ou de créer des fontes dynamiques, mais de faciliter l'adaptation et la modification de fontes par les designers graphiques. Une approche à mi-distance des outils professionnels actuels permettant d'automatiser une partie des tâches de la production de caractères par les designers typographiques et d'un outillage destiné à l'utilisation et l'interaction avec typographie dans un contexte de design graphique. Cette approche, si elle ne s'inscrit pas réellement dans la logique d'une typographie dynamique, amènera probablement à la constitution d'outils qui permettront d'aller dans ce sens.

C Décomposer les glyphes en modules



Permettre une certaine automatisation du dessin de caractères passe par sa décomposition et sa description. Cette décomposition a été réalisée très tôt dans l'histoire de la typographie avec la mise en place d'un vocabulaire désignant l'anatomie de la lettre : fut, empattements, œil, etc. Ce vocabulaire fait écho à la matérialité des éléments de la lettre. Il faut dire que la typographie consiste en un assemblage de formes se répétant d'un glyphe à l'autre. Les modules sont donc une manière très pratique de construire un alphabet à l'esthétique homogène rapidement. Ainsi, de nombreux designers typographiques ont eu recours à des modules pour faciliter la création de leurs caractères. On retrouve cette méthode de construction dans la construction des glyphes des éditeurs de fontes contemporains. Le recours aux modules est également utilisé dans de nombreux alphabets non latins pour assembler différentes formes les unes aux autres et créer ainsi des glyphes complexes. C'est notamment le cas du Mandarin (Chinois traditionnel) et de nombreuses autres langues asiatiques.

De nombreux projets ont tiré parti de la force de la construction modulaire. On peut citer parmi les plus fameux les expérimentations d'Oswald Coopers à la fin du xx^e siècle, le lettrage modulaire aux nombreuses variations de Matthew Carter [\[BIO 16\]](#) pour l'identité du Walker Art Center [\[IMG E4\]](#) ou encore la famille de caractères

History de Peter Bilak [IMG E4], proposant de revisiter l'histoire de la typographie en superposant différents empattements et effets d'ombrages à un même fut. On ne peut cependant pas parler de typographie dynamique dans ces projets. En effet, seule la conception y fait intervenir une modulation des formes, celles-ci étant ensuite rendu définitive ou au mieux modifiable manuellement. Ce n'est pas le cas du *Delorme*, caractère créé en 1990 par Christian Delorme et Jacques André. Cette fonte générative dynamique utilise des modules pour faire varier le dessin et créer une forme unique pour chaque glyphe utilisé. Le principe du *Delorme* [IMG E6] s'appuie sur un squelette de modules s'imbriquant les uns sur les autres pour former les lettres. Ce principe permet de faire varier la forme, et donc la chasse des caractères en recourant à des dessins alternatifs ou des ligatures automatiques permises par le prolongement horizontal de modules. L'automatisation du procédé est réalisée à partir du calcul par l'ordinateur de la valeur d'approche entre chaque paire de caractères et la nature de leurs tracés gauche et droit — répartis en quatre catégories : plein, creux, rond et très creux. Ainsi c'est le logiciel de traitement de texte qui calcule lui-même l'interlettrage nécessaire à chaque dessin généré en fonction du contexte. L'automatisation des décisions de la forme des glyphes, de l'apparition ou non de ligatures et de l'approche des caractères se fait en fonction du contexte, amenant parfois à des résultats surprenants pouvant rappeler les ligatures audacieuses que l'on trouve parfois dans les textes calligraphiés. Mais il existe un point commun à toutes les tentatives que je viens de citer. Il s'agit de projets de typographies de titrage, jamais de caractères de labeur. Cela s'explique

simplement par les limites visuelles des modules. La faiblesse d'une construction typographique modulaire se trouve dans la contrainte imposée aux formes.

Si les modules permettent un certain nombre de variations autour d'une même homogénéité visuelle, ils sont extrêmement contraignants visuellement, car très rigides. De plus, si différentes formes peuvent être réalisées à partir d'une même base, toutes ne se valent pas et il n'est pas garanti que chacune d'entre elles cohabitera de manière harmonieuse. Ainsi, l'aspect et donc le confort de lecture d'une telle fonte ne peuvent égaler les standards d'une typographie dessinée plus librement.

J'ajouterais pour dernier exemple un projet jouant avec la modularité, et dont les enjeux s'imbriquent parfaitement avec le point suivant. Il s'agit du *New Alphabet*. Ce caractère de Wim Crowel, pensé comme une réponse à la faible résolution des écrans de télévision des 1970 — bien qu'il n'y fut jamais transposé. En effet, la faible résolution des premiers écrans CRT permettait difficilement l'affichage de glyphes au dessin complexe, les empattements notamment. Il fallait alors redessiner chaque lettre en réinterprétant son dessin dans le contexte d'une grille à 4, 5, 6... carrés de côtés. Pourtant, l'accès au texte a rapidement été essentiel à l'usage de l'outil informatique — ne serait-ce que pour pouvoir décrypter plus facilement le code informatique.

En effet, si dans les premiers temps on se contente de points lumineux permettant l'affichage de données binaires (allumé ou éteint) [IMG B7], la programmation a

rapidement rendu nécessaire l'affichage de lettres. C'est dans ce contexte que furent réalisés à toute hâte les premiers dessins de fontes « bitmaps » par des ingénieurs, à qui l'outil informatique était alors réservé.

À l'époque, tous les écrans n'utilisaient pas tous des pixels de la même taille (résolution) ni de la même forme (parfois des carrés, parfois des rectangles plus ou moins long), et surtout il n'existait pas encore de format de fonte permettant l'utilisation d'un même caractère sur plusieurs supports logiciels. Avec l'apparition des premières interfaces graphiques, l'outil informatique se démocratise (le *Mac* et *Windows 3.1*). L'affichage de la typographie à l'écran fut longtemps résolu par le dessin de fontes bitmaps, pixel par pixel, pour chaque corps optique. Le dessin de chacune de ces fontes bitmaps était alors défini bit par bit, il était donc statique.

Toutes ces fontes — correspondant aux différentes tailles d'affichage — étaient réunies dans un fichier appelé valise de polices. On pouvait également trouver dans ces valises une fonte *PostScript* — format créé par *Adobe* que je détaillerai plus tard — à contours vectoriels, l'ancêtre du format *Opentype*, à utiliser en cas d'impression.

Le travail de réalisation des fontes bitmaps était alors comparable à celui d'un graveur de poinçons quatre siècles plus tôt, c'est-à-dire un travail d'adaptation du dessin des glyphes à l'échelle du corps optique. Tout ce qui était affiché à l'écran devait être minutieusement défini pixel par pixel pour permettre au dessin adapté d'être utilisé selon la taille de corps voulu. En effet comme l'a



| | |
|--|----------------------------|
| Chicago 12 pt | New York 12 pt |
| Chicago 24 pt | NY 36 pt |
| Monaco 9 pt | San Francisco 18 pt |
| Monaco 12 pt | Toronto 12 pt |
|  | Toronto 9 pt |
| Geneva 9 pt (Cairo 18 pt) | Venice 14 pt |
| Geneva 12 pt | <i>Geneva italic</i> |
| <i>Los Angeles 24 pt</i> | Chicago (outline) |
| <i>Los Angeles 12 pt</i> | |



décrit Richard Southall [BIO 11], là où le graveur de poinçons délimite le noir du signe en enlevant de la matière, le dessinateur de fonte bitmap définit la surface de la lettre bit après bit en désignant les pixels qui seront noircis à l’affichage de la fonte. Et bien que ce travail devait être fort laborieux, il faut lui reconnaître une qualité : celle d’obliger les dessinateurs à adapter le dessin à celui du corps du texte. Ainsi on obtenait, tout comme aux premières heures du plomb, un dessin unique à chaque corps optique.

D Décrire la lettre par ses contours

MACHINE A ÉCRIRE "ADLER"

LITHOGRAPHIE TYPOGRAPHIE

APETERIE

GRAVURE

AMÉDÉE OLIVEAU & FILS

MAISON FONDÉE EN 1825



4. 5. 6. 7. & 8

Quai de la Douane

à Rue de la Douane

15

BORDEAUX

Seuls
Concessionnaires
de l'Autocopiste noir
& de l'Autocopiste
photographique

Photographie

FABRIQUE DE REGISTRES

reliés par le Système Workman

breveté en tous pays

EXPORTATION

Seuls Concessionnaires pour la Gironde

PAPIERS de TOUTES SORTES, FOURNITURES de BUREAUX

Téléphone
J 0 212

Assez rapidement, une autre approche de la description du dessin des lettres s'est imposée. On trouve l'approche du dessin par le tracé central de Donald Knuth avec *Metafont* évoqué plus tôt, mais surtout la description par les contours. Celle-ci était déjà utilisée pour la gravure des poinçons via le dessin précédant leur gravure ou encore les dessins de lettrage sur plaques lithographiques. La courbe de Béziérs est la courbe mathématique la plus utilisée dans l'ensemble de l'industrie graphique. Il existe aujourd'hui de nombreux autres types de courbes (Ikarus, cubique, quadratique [ANN16] [ANN17]), chacune a leurs caractéristiques propres. Mais toutes ont en commun le fait de permettre la description d'une forme par un contour fermé délimitant une surface positive (la zone dessinée), d'une surface négative (la zone laissée vierge). Créée dans les années 1950, la courbe de Béziérs est un objet mathématique qui fut inventé par Pierre Béziérs [Bio] dans le cadre de son travail d'ingénieur pour les usines Renault. Jusqu'alors, la description des formes des carrosseries et autres pièces en volumes était réalisée à travers des montages de courbes matérielles analogiques [IMG C13-C14], rendant les formes peu précises, complexes à mettre en place et à partager. Afin de répondre à ce problème, Pierre Béziérs inventa une courbe mathématique basée sur des points d'ancrage manipulables aux coordonnées définies sur un plan orthonormé. Les points formant les courbes sont fixes et peu nombreux et l'algorithme permettant la traduction de ces points en courbe est simple et peu gourmand en ressources de calcul. Bien entendu, Pierre Béziérs n'est pas le seul à avoir travaillé sur ces courbes. Un des autres grands contributeurs à cette invention est Henry de Casteljaou qui traduisit le concept

de la courbe de Béziérs à travers un algorithme : l'algorithme de Casteljau [BIO 3] [ANN15]. Aujourd'hui, cet algorithme est au cœur de tous les logiciels de reproduction, de rendu et de création graphique. Au-delà de l'anecdote technique et de l'intérêt historique qu'elle peut avoir, la courbe de Béziérs offre de nombreuses clefs conceptuelles permettant de comprendre le fonctionnement de la typographie et donc d'imaginer une typographie dynamique. Cela passe notamment par l'appréhension de son fonctionnement mathématique. En effet, l'algorithme de Casteljau se base sur le calcul de l'interpolation des points marquant le début et la fin de la courbe [Voir Annexe]. Le principe mathématique de l'interpolation est simple et intéressant. Il consiste en une formule mathématique simple [1] permettant de calculer la position d'un point entre deux bornes à partir d'un facteur d'interpolation. Appliqué aux valeurs des courbes de Béziérs d'une fonte, l'interpolation permet d'obtenir par le calcul, à partir de deux dessins maitres, une infinité d'instances de dessins inédits s'inscrivant entre les deux bornes, voir au delà (une extrapolation). Une courbe de Béziérs peut ainsi être rendue « dynamique » par la variation son facteur d'interpolation. C'est le format *PostScript de Type 1* (PST1) qui permet la généralisation de cette courbe. Le format *PostScript de Type 1* (PST1) a été créé et distribué en 1991 par *Adobe*, en même temps que le *PostScript de Type 3* (PST3), version allégée, mais au code ouvert du PST1 [ANN8]. En construisant son format de fonte sur un langage informatique, *Adobe* permet son évolution et son

[1] $y_1 = f(x_1 - x_2) + y_2$ avec f le facteur d'interpolation (x_1, y_1), les coordonnées du point A et (x_2, y_2) les coordonnées du point B.

universalité. Ainsi au fur et à mesure de son histoire le format PST1 s'est enrichi de fonctionnalités s'ajoutant petit à petit à la base du format original. À la différence du PST3, le PST1 est ensuite encodé en binaire, rendant ainsi le fichier illisible et difficilement modifiable par les utilisateurs en dehors d'un éditeur de fonte. Cette particularité freina longtemps la possibilité d'expérimenter et de faire varier le dessin des glyphes à partir du fichier fonte.

La typographie digitale a vu naître de nombreux formats et concepts de description et d’affichage des formes à l’écran. Aujourd’hui la plupart de ces formats ont disparu, bien que certains sont encore employés par des communautés spécifiques. Le format PST1 a résolu le problème de la résolution d’impression des caractères sur le support papier en permettant leur interprétation en adéquation avec le format de la page. Il n’a cependant pas permis une bonne interprétation de ses courbes à l’écran.

Avant l’avènement des écrans à haute résolution [1] le problème de l’interprétation en temps réel des courbes en image bitmap menait la plupart du temps à des aberrations optiques. C’est ce qui força les designers à devoir recourir à des fontes bitmaps. C’est dans ce contexte que fut créé le format *Truetype*. À l’époque de son invention, la résolution des écrans était alors majoritairement inférieure à 72 dpi, laissant ainsi les pixels largement apparents, et le dessin de la typographie gravement altéré, voire illisible. À la différence du PST1 qui utilise des courbes cubiques, dont elle a acheté le brevet à Pierre Béziérs, le format *TrueType* (TTF), créé par Apple en réponse au *PostScript* d’Adobe [ANN7], utilise des courbes

[1] La définition est une dimension informatique, elle nous donne le nombre de pixels en largeur et hauteur. À la différence de la résolution qui représente le nombre de pixels affichés par pouce (la densité de pixels exprimée en Dpi ou Ppp).

quadratiques [W13, p.110]. L'utilisation de ces courbes permet notamment la réduction du nombre de points d'ancrage et donc de la réduction de la taille des fichiers fontes, mais surtout une meilleure disposition des points permettant une meilleure interprétation des courbes dans la grille de pixels et donc une meilleure gestion du crénelage et l'adaptation des contours à la résolution des écrans. Pour réaliser cela, le format *Truetype* fait appel à la programmation de règles internes à chaque fonte permettant de guider le mouvement des points d'ancrage selon les échelles optiques : le *hinting*.

Contrairement à des projets tels que le *Beowolf* ou le *Punk* évoqué précédemment on n'obtient pas une variation de la forme des glyphes les uns par rapport aux autres, mais une réinterprétation de leur dessin dans le cadre de la grille de pixels. Cette fonction répond donc en partie au problème de correction optique dans le cadre de l'affichage sur écran, un problème jusqu'alors compensé manuellement par les designers typographiques. On obtient ainsi, de la même façon que pour les fontes bitmaps, un rendu différent de la courbe à chaque résolution d'utilisation de la fonte. Ainsi la typographie que nous utilisons au quotidien sur écran voit déjà son dessin varier sans que nous en ayons conscience.

Cette étape dans le processus d'affichage me semble intéressante, car elle marque le passage d'une solution manuelle — la création de fontes bitmaps — permettant l'affichage de la typographie à l'écran, à sa résolution par la programmation. La description des glyphes ne se trouve donc plus fractionné en différents formats spécia-

lisés, mais unifié par une même description vectorielle. Cette description est appuyée par des instructions (le hinting) permettant l'adaptation du dessin au contexte de résolution d'affichage.

Aujourd'hui toutes ses technologies sont réunies en un seul format, l'*Opentype* (OTF). Un fichier *Opentype* peut-être de deux natures : .otf ou .ttf. OTF désigne l'Open Type Format, héritier du *PostScript de Type 1* d'Adobe. Le TTF, lui, désigne le *True Type Format* de Microsoft et Apple, dont il est aussi une évolution directe.

Si l'*Opentype* nous intéresse ici, c'est parce que c'est un format de fonte dit « intelligent » capable de substituer un glyphe par un autre selon le contexte dans lequel il se trouve. Il permet de simuler une composition variée du texte, allant même jusqu'à pouvoir afficher une ligne de texte dans laquelle aucun glyphe ne sera utilisé deux fois. Cependant, on ne peut pas parler ici d'une fonte dynamique. En effet, l'*Opentype* ne fait que reproduire, de façon automatisée, ce qui était possible à l'époque du plomb. En outre, pour des raisons de performance, la substitution des glyphes ne peut être réalisée qu'à l'échelle d'une ligne et non d'un paragraphe. La variation des glyphes n'est donc pas envisageable sur un texte complet. [ANN18] Enfin, tout comme les caractères alternatifs de la typographie au plomb, les glyphes se substituant les uns aux autres doivent tous avoir été dessinés préalablement, limitant ainsi très largement le nombre de variations. Ce principe de substitution mime une solution déjà mise en œuvre par Gutenberg lui-même. Lorsqu'il composa la Bible à quarante-deux lignes, il utilisa de nombreux carac-

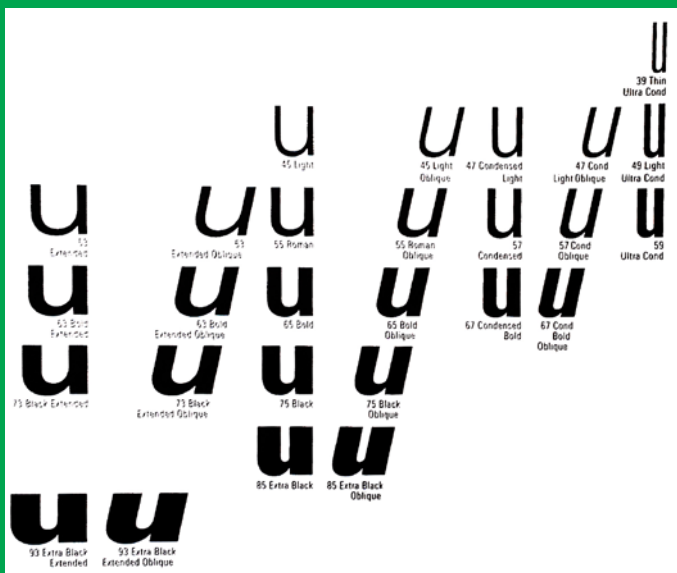
tères alternatifs afin d'imiter l'écriture calligraphique et d'équilibrer la longueur de ses lignes. [w41] Les fichiers fontes ont d'ailleurs, dès les premières heures de la typographie numérique, adapté le concept de casse en le traduisant sous la forme d'une base de données interne au fichier dans laquelle viennent se ranger tous les glyphes et leurs caractéristiques (tracé, espacement) [B2]. Les questions soulevées par la gestion informatique du texte via les fontes numériques sont presque identiques à celles de la gestion physique du texte via les caractères typographiques en plomb. Réglage de la longueur de la ligne, espacement entre les caractères, utilisation de ligatures ou de caractères alternatifs... autant d'opérations qu'il faut encore effectuer aujourd'hui dans l'aménagement de la relation qu'entretient le texte avec l'espace dans lequel il s'inscrit.

Avec le format *Opentype*, il devient possible d'utiliser des courbes cubique ou quadratique, d'adapter l'affichage et le dessin des glyphes à leur résolution d'affichage, tout en pouvant jouer de substitutions contextuelles des glyphes afin d'enrichir la composition visuelle du texte. L'impression et l'affichage d'un caractère à une taille connue sont donc garantis d'être de bonne qualité. Il reste malheureusement impossible de faire varier le dessin des glyphes. Or, je l'évoquais en introduction, aujourd'hui il existe une presque infinité de formats d'écrans, et si les contenus sont conçus pour pouvoir s'adapter aux différents formats, la forme des glyphes se trouve elle dans l'impossibilité de s'adapter aux différentes échelles qui lui sont imposées. En effet, un écran ayant une haute définition — c'est-à-dire un grand nombre de pixels — peut

avoir une basse résolution — des pixels de grande taille. Dès lors il devient impossible de distinguer un écran mesurant quelques centimètres d'un écran mesurant quelques dizaines de mètres à partir de ces deux valeurs. Or, on n'aura pas le même rapport avec une lettre d'un mètre qu'avec un texte de quelques millimètres de hauteur. Cet exemple exagéré s'applique également à une échelle moindre, et dans une telle configuration le hinting ne suffit plus. Seul l'emploi d'un dessin de caractère au dessin optique adapté pourra permettre l'affichage du texte dans des conditions optimales.

Pour pallier cela, la typographie doit devenir dynamique et permettre à son dessin de voir sa forme varier. Le format *Opentype* ne le permet pas, cependant la création d'un environnement actif encadrant le réglage de la typographie ne permettrait-il pas de faire varier la forme des glyphes en fonction de leur taille, de leur environnement ou de la volonté de leur utilisateur ?

F Un dessin multiple



Pour permettre la modification d'un tracé tout en en gardant le contrôle, il est possible de faire appel à l'interpolation. L'interpolation est un algorithme permettant le calcul des valeurs intermédiaire entre deux bornes. Cet outil mathématique est utilisé aujourd'hui par les designers typographiques lors la conception de familles de caractères complexes. Utilisé sur les points d'ancrage des glyphes et leurs valeurs métriques, il devient possible, en faisant varier leur facteur d'interpolation, d'obtenir un caractère régulier à partir d'un caractère léger et d'un gras aux dessins compatibles. On pourra même obtenir un très gras en poussant le facteur d'interpolation au-delà de ses bornes. Il devient ainsi possible d'obtenir une infinité de variations du dessin d'un même glyphe. Ce principe, simple, mais extrêmement puissant permet ainsi aux designers typographiques de créer rapidement des « super familles » (des familles contenant un très grand nombre de déclinaisons) [IMG E1 - 4]. Bien entendu, pour dessiner des formes interpolables les unes avec les autres, un certain de nombre de contraintes est à respecter lors de leur création. Ainsi les deux formes extrêmes devront voir certaines de leurs caractéristiques de base être synchronisées (nombre de point, sens de rotation des courbes, nombre de formes, etc.). Cet important travail préparatoire des fichiers fontes vient s'ajouter à la création en elle-même mais à la différence des techniques précédentes, l'interpolation se traduit par assez peu de contraintes formelles. Le travail

de préinterpolation des glyphes a pour effet de rendre les formes des glyphes homogènes au sein d'une même famille, là où des variations stylistiques auraient pu être réalisées. Cette homogénéité peut devenir lassante, et risquer d'appauvrir la langage typographique. Mais l'accès à des variantes stylistiques d'un même glyphe peut en partie résoudre ce problème. Il apparaît clairement que les contraintes propres à l'interpolation n'ont rien à voir avec les contraintes formelles jusqu'alors subies dans le cas de la typographie au plomb, la photocomposition ou le dessin de fontes bitmaps. Le principe d'interpolation fut donc très logiquement exploité dès 1991 afin de créer une extension du format PST1 permettant l'interpolation de plusieurs dessins fixes dits « maitres », contenus dans un même fichier fonte. Ce nouveau format appelé *Fonte Multiple Master* fut un échec commercial. Pourtant l'idée d'une fonte ne contenant plus un, mais plusieurs tracés et permettant un « morphing » entre chacun d'entre eux semblait, et semble toujours une excellente idée. Imaginez-vous pouvoir afficher ou imprimer une famille de caractère complète et l'adapter à son contexte d'utilisation sans avoir à redessiner ou à altérer son dessin. Tout le potentiel d'une typographie dynamique était là. Cependant la faiblesse du format *Multiple Master Fonte* (MMF) ne réside pas dans son utilisation de l'interpolation, mais dans le contexte dans lequel il propose son utilisation. Créés par *Adobe*, les MMF sont arrivés en pleine guerre des formats. À cette époque il était impossible d'utiliser une fonte *Mac* sur *Windows* et vice versa. Pour contourner cette limitation et permettre l'utilisation des fonctionnalités d'interpolation des MMF, *Adobe* créa *Adobe Type Manager* (ATM), un programme,

gratuit mais limité à la création d'instances de fontes à partir des MMF, c'est-à-dire la génération d'une version fixe d'un état interpolé au format PST1. Dès lors l'interpolation perdait tout intérêt, et la fonte tout dynamisme. Les seules fontes qui furent commercialisées dans ce format furent donc réalisées par *Adobe* et le format disparu assez rapidement sans jamais avoir été réellement adopté et exploité à la mesure de son potentiel.

Quelques projets notables furent néanmoins réalisés par *Adobe*. On peut citer parmi eux l'*Adobe Penumbra* [\[IMG E8\]](#) créé en 1994 par Lance Hidy. Une famille de caractère interpolant un dessin sans-sérif géométrique avec son équivalent à empattement et faisant ainsi le rapprochement formel entre les formes modernistes géométriques et les canons Antiques telle que le Trajan. Je citerai également le développement d'une version du Minion permettant une interpolation des glyphes sur la base de cinq dessins optiques permettant un réglage visuel optimal de ce caractère de labueur de la légende au titrage en passant par le texte courant.

Apple est également à l'origine d'un troisième format, le AAT (*Apple Advanced Typography*), anciennement nommé *Truetype GX*. Un format qui ne vous dira certainement rien, mais sur lequel nous reviendrons, tant son influence fut importante sur la typographie telle que nous la connaissons. Développé à partir du format TTF de Apple, le *Truetype GX* (TTGX) fut précurseur dans les fonctionnalités qu'il permettait aux utilisateurs. Une grande partie d'entre elles offrent aujourd'hui encore une plus grande finesse typographique que ce que permet

l'*Opentype*. Le TTGX reprend — à l'instar des MMF — le principe de l'interpolation pour faire varier la forme des glyphes dans différents axes prédéfinis. Mais à la différence des MMF, il ne nécessite pas l'installation d'un logiciel et permet l'interpolation en direct dans les applications qui utilisent la technologie de rendu de texte Quickdraw GX. Cependant, cette technologie n'étant proposée que sur Mac OS, le TTGX ne fut jamais adopté par les designers typographiques. De plus, il ne fut jamais implémenté sur d'autres logiciels que ceux conçus par Mac, le rendant ainsi invisible aux designers, travaillant sur des logiciels professionnels tels qu'*Adobe inDesign* ou *QuarkXPress*. D'ailleurs aucun éditeur de fonte ne prit en charge ce format, encore utilisé aujourd'hui par *Mac OS X* et *iOS*. Pourtant ce dernier va très loin dans la gestion de la typographie. Il introduit le principe de substitution contextuelle des caractères et de jeu de caractères stylistiques, ainsi que la possibilité d'utiliser des chiffres elzévirien, tabulaire, des fractions verticales, horizontales, etc. Des fonctions qui seront par la suite reprises par le format *Opentype*, créé par Microsoft et *Adobe*. Le format TTGX, renommé entre temps *Apple Advanced Typography* (AAT) proposait également, couplé à son moteur de rendu QuickDraw GX, une gestion des paragraphes. Une fonctionnalité qui ne l'a probablement pas aidé à être implémenté dans les logiciels professionnels de mise en page qui font de cette fonctionnalité un de leurs arguments de vente. [v15] Ainsi, bien que répondant à la plupart des caractéristiques d'une typographie dynamique, le format AAT est resté marginal et aucune fonte utilisant cette technologie n'a été distribuée, sinon celles créées pour son lancement. Pourtant des curseurs permettant de faire

varier les facteurs d'interpolation et donc les formes des glyphes furent créés et mis à disposition des développeurs dans un logiciel de démonstration précompilé à l'intention — Wordtext accessible via xCode sur *Mac OS* dans ses versions 10.3 à 10.4. Cette proto-interface reste à ce jour le seul moyen visuel d'utiliser les fonctions de variation du format AAT. Les fonctionnalités du format restent cependant accessibles via le code et pourraient être implémentées dans de nouveaux logiciels. L'intérêt d'une telle mise en place reste cependant faible, le format AAT n'étant pas pris en charge par les éditeurs de fontes et limitée dans son utilisation au système *Mac OS X*. Aujourd'hui ce format demeure à l'état de vestige dans nos systèmes d'exploitation à travers la fonte Skia, distribué en tant que fonte système dans tous les *Mac*. Cette famille de caractère créée en 1993 par Matthew Carter et David Berlow sur une commande d'Apple pour le lancement de Quickdraw GX et du format TTGX permet d'entrevoir tout le potentiel du format. Elle est notamment connue des designers et amateurs de littérature français pour son utilisation en corps normal foisonnant de ligatures insolites sur les couvertures des éditions Cent pages par le designer Philippe Millot. [1] Des ligatures incluses dès l'origine directement dans le fichier fonte via les fonctionnalités offertes par le format. Cette famille de caractère [1] est interpolé en grasse et en largeur. À cela s'ajoute le grand nombre de caractères alternatifs contextuels et stylistiques permis par les fonctionnalités du format TTGX que j'évoquais précédemment. Cette fonte n'est aujourd'hui utilisée que très partiellement,

[1] Skia se traduit du grec par « l'ombre ». Elle s'inscrit en référence aux premiers alphabets grecs.

les logiciels de mise en page professionnels tels qu'*Adobe*, *QuarkXPress* ou même *Scribus* n'utilisant que son tracé de base (normal), comme s'il s'agissait d'une simple fonte *TrueType*, sans permettre l'accès aux fonctionnalités du format TTGX.

Malgré ces quelques percées dans l'utilisation de la typographie, l'interpolation reste aujourd'hui limitée à une utilisation dans le cadre de la création typographique. Des logiciels tels que *Superpolator* permettent l'utilisation de l'interpolation pour automatiser le travail de déclinaison des caractères d'une famille typographique. Le projet *Prototype* s'inscrit également dans cette démarche tout en la poussant plus loin. Amorcé en 2014, il utilise le principe de l'interpolation appliqué à des modules qui composent les glyphes. Il devient alors aisé d'accentuer ou de réduire la taille et la forme empattements, du fut, de la goutte, etc. Cet assemblage de formes définies par leurs contours permet de générer des caractères typographiques aux formes différentes assez facilement en intervenant sur un ou plusieurs paramètres prédéfinis. Cependant, tout comme *Metapolorator*, ce projet s'appuie sur la base de squelettes modèles pour définir les caractéristiques et facteurs de variation des dessins sur la base. Ainsi ces deux projets se trouvent — pour le moment — à la rencontre entre un éditeur typographique et un outil de variation paramétrique [IMG D13]. Au regard de l'histoire, il apparaît que la technique modèle notre relation avec le matériau typographique, et réciproquement. Il est maintenant clair que le passage de la typographie du modèle statique à celui de son dynamisme a déjà eu lieu à de nombreuses reprises et aura lieu à nouveau. L'une des clefs qui

fera, ou non, de ce passage un état durable sera le choix d'une nouvelle relation au texte et au matériau typographique dans sa conception et son utilisation. Vers quel modèle désirons-nous aller? Continuer notre utilisation actuelle à travers des interfaces graphiques simulant le rendu final? Ou une approche par le code, déjà à l'œuvre sur Web? De même, allons-nous vers la création d'un format inédit ou d'un appareil technique venant renforcer les formats actuels?

3 Évoluer dans l'espace

A Vers quelle typographie dynamique ?

Le potentiel dynamique des formats tels que les MMF ou l'AAT (toujours utilisé et supporté par *Mac OS X*) laisse rêveur. Pourtant aucun de ces formats n'a réussi à s'étendre au grand public. Il me semble que poser la question d'une typographie dynamique passe par le questionnement de l'échec de ces formats et force à se demander si la solution passe réellement par un nouveau format de fonte. L'incapacité des MMF d'*Adobe* et du AAT de *Apple* à se démocratiser est passée par de nombreux aspects. [1] Lors d'un échange sur le forum *TypeDrawers*, j'ai pu poser mes questions aux acteurs de ces technologies passées et évoquer mes hypothèses sur les raisons de l'échec de ces technologies [voir Document en Annexe sur les raisons de l'échec des MMF et du AAT].

Il ressort clairement de cet échange que le manque de support de l'environnement logiciel permettant l'utilisation de la typographie digitale (en particulier concernant les logiciels professionnels de mise en page tels qu'*Adobe*) fut l'un des handicaps majeurs de formats tels que les MMF et le TTGX. De ce point de vue la fermeture des logiciels et l'absence d'alternative libre furent, à l'époque, un frein important. Mais plus encore

[1] Dans une présentation intitulé *Why OpenType Succeeded Where GX & MM Didn't* [v15] réalisée lors de l'ISType (Istanbul Type Conference) le 13 juin 2015, Thomas Phinney tente de faire le point sur les raisons de l'échec de ces technologies et essaye de comprendre ce qui forgea à contraitrio le succès de l'Opentype.

qu'un manque d'investissement des formats par les grandes firmes logicielles de l'industrie graphique, il me semble que c'est la dépendance des designers aux interfaces utilisateurs qui creusa le tombeau de ces formats. En effet, nous l'avons vu, les fonctions offertes par ces technologies nécessitent une action de la part de l'utilisateur (faire varier le facteur d'interpolation). Or les logiciels de composition et de mise en page de texte tels que la suite *Adobe* n'ont jamais mis à disposition d'interface utilisateur permettant cela. Seul *Illustrator 10* pour les MMF et une application de démonstration réservée aux développeurs [1] pour les AAT mit à disposition des curseurs permettant de faire varier les paramètres des fontes. Dès lors, il était évident que, ne pouvant pas — ou difficilement — accéder aux fonctionnalités dynamiques les utilisateurs n'ont pu que difficilement en saisir l'intérêt, et donc les adopter. Le même problème freina longtemps l'utilisation des fonctionnalités *Opentype*. Le menu permettant de les utiliser dans les logiciels *Adobe* fut longtemps — jusqu'à la suite *Adobe Creative Cloud 2015*, soit 20 ans après le lancement du format — limité à un sous onglet d'un menu auquel on accédait à travers une petite icône non explicite dans une fenêtre permettant le choix et le réglage de la police de caractère... La dépendance des designers aux logiciels de la suite *Adobe* et aux interfaces utilisateurs mena même jusqu'à une pétition [w43] en octobre 2014 pour demander à la compagnie américaine la création d'une véritable interface permettant l'utilisation des capacités du format qu'elle avait créé 20 ans plus tôt. Il faut dire que l'interface jusqu'alors existante avait été créée à l'initiative d'un employé zélé d'*Adobe*,

[1] [Wordtext sur Mac OS X 10.3 et 10.4](#)

ayant eu l'idée d'implémenter cette fonctionnalité. Cette situation absurde est décrite par Dave Crossland lors d'une discussion sur l'importance des logiciels libres dans la création typographique [w44]. Les formats propriétaires et fermés tiennent donc une part de responsabilité dans l'échec des MMF et du format AAT.

B Dépendance aux interfaces utilisateurs

À l'origine de notre dépendance aux interfaces utilisateurs, on trouve la philosophie du WYSWYG [1] développé au Xerox PARC (Palo Alto Research Center) aux États-Unis dans les années 1970 puis repris par Apple dans les années 80 avec la sortie du premier *Mac* grand public. Né de la volonté de faciliter l'accès et l'utilisation de l'outil informatique, le WYSWYG fait interagir l'utilisateur avec les documents en simulant visuellement leur état final d'impression ou d'affichage. Le code se trouve donc caché derrière des interfaces permettant l'utilisation de fonctions préimplémentées, directement traduites visuellement à l'écran et montrant le résultat en temps réel. J'ai la conviction que l'évolution de la typographie vers une forme dynamique passe par la redéfinition progressive de notre interaction avec le texte. Cette évolution a déjà commencé sur le Web, où les designers manipulent le texte directement par le code. On quitte alors le WYSWYG pour aller vers le WYSWYM [2]. La nature du texte avec laquelle le designer travaille n'est alors plus celle d'un texte formaté visuellement, mais d'un texte brut accompagné de règles le structurant par le biais de code informatique. D'une programmation visuelle implicite passant par une interface utilisateur, on passe à une définition explicite des caractéristiques du

[1] *What You See What You Get*, c'est-à-dire ce que vous voyez est ce que vous obtenez ou Tel écran, tel écrit

[2] *What You See is What You Mean*, c'est-à-dire Ce que vous voyez est ce que vous voulez dire

texte. Cela ne veut pas dire pour autant que le designer travaille en aveugle sans avoir de retour visuel sur les résultats de son code, ce qui était l'un des principaux défauts de *Metafont*. Des éditeurs de code permettent aujourd'hui le live coding (le code en temps réel). Ceux-ci génèrent en temps réel le résultat du code en train d'être créé, permettant ainsi un travail visuel aussi précis que le WYSWYG, sans aucune ambiguïté. Le programme peut ainsi prendre en compte les différents formatages du texte selon différents contextes définis par le designer. Cette programmation du formatage du texte, jusqu'alors dépendante des possibilités offertes par les interfaces logicielles, devient accessible sans autre contrainte que celles que le designer s'impose.

Les logiciels sont des produits commerciaux de masse, ils ne sont donc pas créés pour des individualités, mais pour un groupe cible avec des souhaits et des demandes hypothétiques. Si en pratique ce groupe cible s'avère avoir des désirs inassouvis, des fonctionnalités supplémentaires sont ajoutées avec une mise à jour, mais cela seulement dans le cas d'un intérêt suffisant. [...] Dans le passé vous pouviez réussir si vous pouviez faire ce que vous aviez à l'esprit avec vos propres mains. Aujourd'hui le danger est que les designers ne soient efficaces que si leur créativité n'excède pas le potentiel de leurs ordinateurs : si le monde des idées rentre dans le monde des outils. Bien sûr, à long terme, cela conduit à une monotone uniformité visuelle conduite par l'ordinateur. [w39] Aujourd'hui des applications telles que Drawbot [1] pro-

[1] Créé par Letterror et Frederik Berlaen et utilisant le langage Python

posent un environnement dédié dans lequel utiliser le code pour créer des documents imprimés. Une autre approche est utilisée par la librairie *HTML2Print*, développée par Open Source Publishing. Celle-ci essaye, à travers un ensemble d'outils préconçu, d'adapter l'utilisation des technologies Web au design de documents à vocation imprimé. Une approche unifiant les outils de création de documents autour des langages Web. Le format papier devient ainsi un état particulier s'inscrivant dans la logique de l'adaptation de la page aux multiples formats des dispositifs d'affichage. La question du code comme outil pose également la question de son accès et de sa compréhension par les utilisateurs. Sans trop m'étendre sur ce sujet qui dépasse largement les questions soulevées dans ce mémoire, il me semble intéressant de souligner l'importance que peut avoir la forme du langage permettant l'interaction avec le texte. Un langage informatique permet d'intervenir de manière active sur des éléments, et donc d'automatiser une partie de l'écriture ou de la réécriture du code lui-même selon des variables définies (taille du document, résolution, météo, date, etc.). L'assimilation d'un langage informatique demande un apprentissage, et donc du temps de la part d'un utilisateur. Afin de rendre cet apprentissage le plus simple possible et de faciliter son assimilation, le Xerox PARC [2] développa à partir de 1972 un langage de programmation dont l'idée était de s'approcher au maximum du langage naturel : le *SmallTalk* [3]. Par la suite de nombreux langages informatiques s'en sont inspirés. C'est notamment

[2] Palo Alto Research Center



[3] Conçu par Alan Kay, Dan Ingals, Ted Kaehler et Adele Goldberg

le cas du langage *Python*, un langage très présent dans le monde de la typographie en raison de sa simplicité et de sa concision permettant la création rapide de petits scripts permettant l'automatisation de tâches répétitives. Il est notamment utilisé dans le projet *Drawbot* et dans la plupart des éditeurs de fontes. Avec l'avènement du Web, l'accès au code tend à se démocratiser peu à peu et de plus en plus de designers travaillent directement avec cet outil. Que ce soit pour la réalisation d'interface Web, d'applications ou d'objets imprimés, la prise de conscience du code sous-jacent aux interfaces visuelles permet d'envisager une transition vers un mode de gestion du texte plus conscient donc plus malléable, précis et personnel.

C Construire des outils sur mesure

The Desperate **Chainsaws**
19: Open Mike Night* (A)
20: NoMoreCurves (UK)
+ Bassment (Amsterdam)
21: Urban Death Ray (D)
Assinine Megaturbidites
22. Sedimäntary Bodeez
23. De@thSchr!ft (USA)
LastPartay at this venue!
>**PLZ**<Add2Shoppingcart?
And Tonight Only: Dädadaist Pønks

Le développement d'une relation personnelle à la gestion du texte et à la typographie passe par l'utilisation d'outils adaptés, personnalisés, voir créés sur mesure.

Bien qu'il soit certain que l'histoire des formats de fonte ne soit pas terminée, nous ne pouvons aujourd'hui plus penser la typographie sans prendre en compte l'état actuel de la technologie utilisé par les designers. La solution n'est donc peut-être plus seulement à trouver dans la création d'un nouveau format, mais plutôt dans son environnement d'utilisation. En effet, le format *Opentype* est un format ayant fait ses preuves, et il restera probablement de nombreuses années encore le format standard utilisé par les designers et typographes. De plus, il permet d'ores et déjà de simuler une partie des fonctionnalités attendues dans une typographie dynamique. Le *Beowolf*  de Letterror y a été porté en utilisant les fonctionnalités de substitution contextuelles pour simuler la variation du dessin de ses glyphes. Des projets tels que le Liza d'Underware  [Ref] et sa simulation de l'encrage d'un pinceau pour faire varier le dessin des lettres (fonctionnalité Out of Ink) montrent parfaitement la puissance, mais également les limites de ce format. Lors de l'Atypi 2015 à Sao Paulo, une table ronde fut organisé afin de prendre connaissance des besoins et envies des designers et autres professionnels de la typographie pour le futur du format. [\[W23, P.111\]](#) Parmi les nouvelles spécifications à l'étude, on trouve un réin-

vestissement des caractéristiques de variation du format AAT dans la nouvelle version du format *Opentype*. Bien entendu, la création et la distribution de ce format prendront du temps et rien ne garantit que les variations seront implémentées dans le format final. L'ouverture et la consultation des designers semblent cependant aller dans la direction d'un format dynamique, ouvert et à l'écoute des besoins des utilisateurs, ne s'inscrivant plus dans une logique de compétition de format propriétaires. Mais l'enjeu autour de la typographie dynamique me semble largement dépasser celui de la création d'un format adapté. On trouve aujourd'hui des expérimentations typographiques utilisant des outils dérivés d'autres usages, tels que le Web ou le scripting.

Je citerai deux des plus importantes et prometteuses : *plumin.js* [1] et *opentype.js* [2]. Ces deux bibliothèques ont déjà commencé à pousser les possibilités offertes par le format *Opentype* en créant un environnement logiciel actif permettant par exemple la modification du dessin des glyphes affiché à l'écran, leur animation voir leur interpolation. À l'origine de ces deux projets, on trouve, comme dans le cas du *HTML2Print*, un basculement intéressant du support d'affichage navigateur Web du statut d'outils de consommation de contenu à celui de création avec des projets tels que *Prototypo*, *Metapolator* ou *Glyphr*.

[1] Yannick Mathey et Louis-Rémi Babé
< <https://github.com/byte-foundry/plumin.js> >

[2] Frederick de Blaser
< <https://nodebox.github.io/opentype.js> >

Cela m'amène à évoquer le projet *CSS Sans* [3] [IMGX, P.XXX]. Un projet expérimental de typographie utilisant le langage CSS3 — troisième version du Cascading Style Sheets, langage de description de style des éléments HTML — comme langage de description des glyphes. Extrêmement limité dans ses formes par les possibilités du langage lui-même, ce projet n'a que très peu d'intérêt d'un point de vue pratique. Il est néanmoins intéressant dans sa démarche d'intégration de la typographie comme élément constitutif de la page Web au même titre que les images et les blocs, et non plus comme élément externe à celle-ci. Cette approche inclusive de la typographie comme élément de la page a pour effet sa possible mise à l'échelle et l'évolution de sa forme avec la page elle-même. Bien qu'il permette de composer du texte, ce projet ne s'intéresse par réellement à la composition de texte, mais plutôt au titrage, voir au lettrage. Le lettrage est un travail signalétique réalisé à l'échelle du signe et non plus du texte. Très présent sur les supports imprimés dans lesquels il est caractérisé par sa capacité à s'adapter à son environnement, le lettrage est assez peu utilisé sur les supports numériques où, l'environnement pouvant changer de forme, il perd une grande partie de sa force. Le projet *Digital lettering* de Lettererror propose de répondre à ce problème en permettant la création de lettrage s'adaptant à son espace en utilisant le format SVG (un format de description vectorielle basé sur le XML, tout comme le HTML). Couplé à un script reprenant le principe de l'interpolation, le lettrage peut alors s'animer et se déformer aux bons vouloir de son créateur et de son utili-

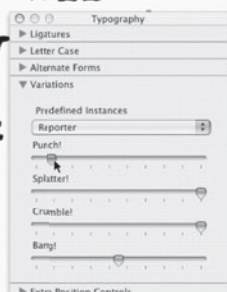
[3] [CSS Sans](http://yusugomori.com/projects/css-sans/fonts)
< yusugomori.com/projects/css-sans/fonts >

sateur. Ce projet reprend, dans sa mise en place, la même idée que le CSS Sans : faire des glyphes des éléments à part entière de la page. Il va cependant plus loin. Ici il n'existe aucune limite visuelle aux formes, dont le rendu est équivalent à celui d'une fonte classique, puisqu'elles sont décrites par des courbes de Béziérs. Cependant, s'agissant ici véritablement de lettrage et non pas de typographie, il est impossible de composer du texte. Le dessin forme un tout indissociable, telle une ligature [IMG E12].

Ces différentes utilisations du code démontrent que la typographie dynamique peut exister sans format la prenant en charge. Chaque besoin requiert un outillage spécifique et l'utilisation active de la typographie par le code peut répondre à ce problème. Au-delà des nombreuses considérations techniques et pratiques soulevées jusqu'ici, de nombreuses questions restent en suspens.

D Les enjeux de la typographie dynamique

Jam by Erik
van Biekländ
TrueType GX
Responsive
Font 1991
Awesome!



Tout au long de ce mémoire, l'un des principaux enjeux de la typographie dynamique a semblé être la résolution du problème de la mise à l'échelle optique des caractères.

Nous avons vu que quelques projets historiques tels que le *Minion MM* avaient atteint cet objectif, malheureusement ces derniers ne sont plus utilisables dans les conditions logicielles actuelles. Récemment (2013), Matthew Carter a participé à la réalisation d'un projet, en collaboration avec la fonderie de caractères Tiro Typeworks, avec pour ambition la mise à l'échelle optique automatique du dessin des glyphes à l'écran : le Sitka (faut-il y voir un clin d'œil au *Skia* ?), commandé par Microsoft pour son système d'exploitation *Windows 8.1*. La solution retenue fut bien plus simple que celles envisagées dans les exemples cités précédemment. Ici, la forme des glyphes ne varie pas de manière mathématique, le format substitue un dessin à un autre. Plusieurs graisses optiques ont donc été réalisées et intégrées au fichier fonte. Ces dernières sont ensuite simplement sélectionnées par les applications en fonction de leur contexte d'affichage [w51]. Ce projet, moins révolutionnaire dans son approche que le *Truetype GX*, montre néanmoins que l'intérêt et la recherche de solutions suscités par la question de l'adaptation du corps optique et de la variation du dessin typographique sont loin d'être terminés. Il reste tout de même étonnant de devoir multiplier les instances de fontes plutôt que d'utiliser la puis-

sance de calcul — aujourd’hui largement suffisante — permettant de les calculer à partir d’un nombre minimum de tracés. Si cet état ne pose pas réellement de problème pour une utilisation hors ligne, le poids des fichiers fontes reste un frein à l’utilisation de multiples variations sur le Web. Dans ce cas, une typographie dynamique serait plus flexible et plus légère. L’enjeu d’une typographie au dessin dynamique ne se trouve pas seulement dans la possibilité d’une typographie plus précise optiquement, plus efficace techniquement, ou plus fine visuellement. Son potentiel dépasse de loin des considérations telles que la gestion automatisée du corps optique. Il permettrait — à mon avis — un renouveau sensible dans l’utilisation et la manière de penser la typographie digitale, de même que dans sa conception et son industrie. J’en veux pour preuve un des rares projets qui fut réalisé avec le format *TrueType GX* [1]. La fonte Jam réalisée par Erik Van Blockland dans le cadre d’une commande d’Apple pour *FontBureau*, utilise les fonctionnalités de variation du dessin de TT GX pour simuler la frappe d’une *machine à écrire*. Au lieu d’utiliser les axes d’interpolation pour faire varier des caractéristiques classiques telles que la graisse, la largeur des glyphes ou l’italicité, on trouve quatre axes aux noms évocateurs : *Punch!*, *Splatter!*, *Crumble!* et *Bang!*. *Bang!* permet de faire varier les accidents d’encre jusqu’à remplir l’œil d’une lettre entièrement de noir, *Crumble!* modifie la graisse des brisures dans les glyphes, *Splatter!* change la taille des brisures dans les glyphes. Enfin, *Punch!* permet de nuancer l’encre et la force de frappe, et donc la graisse et l’empatement

[1] Une liste de fontes créées dans ce format se trouve à www.gxfanclub.com/gxfonts.html

du dessin. Le projet *Jam* [IMG E9] montre, en tentant de reproduire l'effet d'une frappe de machine à écrire, qu'il est possible d'imaginer une infinité de variations non conventionnelles. Déconstruction de la lettre, variation du gris optique, hybridation progressive de formes, simulation d'effets (peinture, bombe aérosol, calligraphie plus ou moins achevée...), la variation ouvre un nouveau champ d'expression à la typographie de titrage, voir labeur. Actuellement le format *Opentype* ne permet pas ces variations, pourtant, on trouve des expérimentations et des projets tendant vers ce type d'usage de la typographie. Ainsi le *Trixie*, réalisé par Lettererror, tente de réaliser, dans la mesure des possibilités offertes par l'*Opentype*, de proposer une partie du programme visuel pensé avec le *Jam*. Plus récemment le *Tripper*, fonte de titrage stencil de la fonderie Underware, simule, avec sa version *Rough*, les variations et effets de la peinture à la bombe sur des pochoirs à stencils.

De tels caractères typographiques remettent en question de nombreux aspects, jusqu'alors restés indiscutés dans le processus de création et d'utilisation de la typographie. Jusqu'à aujourd'hui, la typographie a toujours — hormis de brèves tentatives — été distribuée comme un ensemble de glyphes à l'image fixe dessiné par le designer. En proposant une typographie pouvant voir son dessin varier, le designer typographique perd une partie de la maîtrise visuelle sur sa famille de caractère. Il ne peut plus penser ses choix visuels de la même manière. Aujourd'hui le designer typographique choisit les instances de fontes qu'il décide de générer. L'utilisateur se voit ainsi contraint dans son choix de graisse, de largeur

ou d'italique de son caractère. Une contrainte qui peut être vécue comme une frustration, mais aussi comme une manière de délégué le choix du meilleur gris et du meilleur état visuel de la fonte à un spécialiste. Les typographes sont-ils prêts à laisser la possibilité de ce choix aux designers? Et surtout, les designers sont-ils intéressés par une telle finesse dans le choix de leurs réglages typographiques? Après tout, les choix opérés par les designers typographiques ne seraient-ils pas une sorte d'éditorialisation des formes, au même titre qu'un éditeur choisit les textes qu'il propose à ses lecteurs?

Ce n'est pas toujours le cas, mais le plus souvent du matériel est confié au designer graphique, sur lequel il travaille et opère une sélection. Le design n'est habituellement pas une création partant de zéro. [...] Autre chose doit être ajouté au sujet du langage, car l'idée clé, le mot-clé de cette conférence — éditer — se retrouve différemment au sein même des langues. En anglais, sans doute, l'éditeur est celui qui traite le matériel édité, le trie, l'agence, travaille à en réduire la taille ou demande à l'étendre, enfin y met bon ordre. En français il me semble que sous ce terme sont superposées deux notions : celle qui vient d'être évoquée et le fait de publier. En français, un « éditeur » est ce que nous nommons en anglais un « publisher » et je peux supposer que cette signification témoigne d'une persistance d'une idée historique de la publication. C'est l'idée que l'« éditeur » tel qu'il travaillait avant l'avènement des grandes maisons d'édition — tant comme éditeur des textes à proprement parler qu'en tant que celui qui prend soin de la production des objets, mais encore de leur distri-

bution et leur vente. Chez moi ces deux fonctions se combinent, bien que cela devienne moins fréquent et que je ne tiens pas spécialement à recommander d'associer ces deux activités. [1]

On trouve d'ailleurs à ce propos tout un débat sur la manière de penser les instances de graisse à développer. Ainsi Pablo Impallari et Lucas de Groot ont chacun proposé une approche mathématique de la question. Approche qui pose une fois encore la question de l'utilité de devoir créer ses instances manuellement [2], leur choix pouvant être réalisé par le calcul.

La présélection d'instances suggérées par le designer typographique pourrait être une bonne manière de faire le lien éditorial entre le créateur et l'utilisateur.

Bien entendu une telle typographie ne serait pas utilisée par tous à son plein potentiel. Mais que ce soit par le biais d'un atout plastique ou une plus grande précision dans la gestion typographique, je suis persuadé qu'elle serait mise à profit par les designers graphiques.

[1] Robin Kinross, Extrait de conférence donné à l'ÉSAD. Valence en 2013.

[2] < impallari.com/familysteps >

E Qui laisser intervenir sur la forme du texte ?

Actuellement la plupart des licences d'utilisations interdisent la modification des tracés des caractères. Que se passerait-il si un designer décidait d'utiliser une famille de caractères existante dans un environnement logiciel permettant leur interpolation ? Quels seraient alors le statut et la légalité des dessins intermédiaires générés par l'algorithme ? La question se pose particulièrement du fait que les licences de fontes commerciales actuelles ne prévoient pas cet usage. La modification n'étant pas réalisée à l'intérieur même du fichier fonte, mais dans le cache de l'ordinateur, on se trouve dans un flou juridique total. Une autre question se pose : à qui laisser la maîtrise de la forme typographique ? L'auteur d'un texte, son transcripteur, le designer graphique ? Chacun de ces utilisateurs demande une interaction spécifique avec le texte, et donc avec la forme des glyphes.

Les puristes vont dire, « Whoa, calmez le jeu ». Personnellement je suppose que les scribes et les moines copistes ne pensaient pas non plus que Gutenberg avait eu une bonne idée. [1]

Nous l'avons vu, aujourd'hui des proto-outils permettant une utilisation dynamique de la typographie existent.

[1] Allan Haley, Director of Words and Letters, Monotype « The Executive Computer; In the Latest Type Technology, an Echo of Gutenberg, » Peter H. Lewis, « The New York Times, » March 17, 1991.

Cependant ils sont réservés aux designers typographiques. Un écrivain aura un tout autre usage de la typographie dynamique. J'ai déjà évoqué, à travers le projet de Benjamin Gomez permettant une retranscription de l'expressivité de l'écriture dactylographiée par la déformation des glyphes, une des pistes d'usage envisageables de la typographie dynamique dans le contexte de l'écriture. Pourquoi ne pas imaginer un réinvestissement de la forme typographique — au-delà même de son formatage — dans le contexte de la création littéraire, ou même de l'écriture quotidienne (mail, SMS, commentaires, etc.)? Un renouveau expressif de la typographie qui pourrait pallier le manque de nuance dans le ton offert par la typographie, aujourd'hui en partie compensé par l'utilisation de smileys. On pourra bien entendu opposer à cette approche la vision défendue par Béatrice Warde dans *The crystal Goblet* dans lequel elle prône un retrait de la forme typographique pour laisser plus de place au propos et permettre une plus grande clarté de celui-ci.

Un autre usager potentiel de la typographie dynamique est le lecteur. Il existe aujourd'hui déjà sur les dispositifs de lecture à écran (tablettes, liseuses, etc.) la possibilité de régler le texte à sa guise (corps du texte, interlignage et même choix de la fonte). Que les designers soient pour ou contre laisser la possibilité au lecteur d'intervenir sur la forme du texte — tout comme certains designers typographiques pourront s'opposer à laisser l'accès au designer graphique à la modification des réglages du tracé des glyphes — n'est pas la question, ce fait est désormais inévitable. Mieux vaut donc l'encadrer en programmant les interactions du lecteur avec la typographie pour per-

mettre une qualité constante de celle-ci, quels que soient ses réglages. La typographie dynamique rendrait possible la mise à disposition de variations du dessin des glyphes pour les lecteurs dyslexiques ou mal voyants directement dans le même fichier fonte.

Conclusion

La typographie dynamique m'a tout d'abord sauté aux yeux par son absence. Un creux semblait se dessiner en son endroit. Notre environnement textuel, technique, nos usages de la typographie... tout semble converger dans le sens de sa possible — et nécessaire — variabilité. Pourtant jusqu'à aujourd'hui elle semble résister à cet appel.

Nous l'avons vu ensemble, la typographie dynamique a existé et réexistera bientôt à travers une mise à niveau du format *Opentype*. Mais il m'est apparu en écrivant ce texte que penser une typographie dynamique ne se limitait pas à penser un format de fichier ou une mise en action de la forme des glyphes. La mise en mouvement — animé ou non — de la forme d'un texte remet en question la notion même du dessin typographique. Ainsi, les pistes que j'ai pu évoquer ont soulevé un certain nombre de questions qui restent sans réponse : la question juridique, celle du statut du dessin et de sa modification potentielle par l'utilisateur, mais aussi celle de la composition du texte et de son formatage.

La typographie dynamique changera la vision que nous avons d'une fonte et de la composition du texte. Le passage du statisme à une potentielle variation change le paradigme dans lequel s'inscrit le texte. Dès lors, tout son appareillage technique est à repenser. Je suis convaincu que l'utilisation et la compréhension du potentiel dynamique de la typographie passeront par une utilisation

du code informatique associée à un environnement graphique hybride mêlant WYSWYG et WYSWYM qui permettra un usage simplifié, mais conscient des outils typographiques.

Reste ensuite à savoir ce qu'en feront les designers graphiques et typographiques. J'aspire à voir le potentiel de cette typographie exploité au-delà des variations de son corps optique ou son adaptation aux différents formats d'écrans. J'espère avoir partagé, à travers ce mémoire, l'enjeu plastique soulevé par la typographie dynamique. Un enjeu qui permettra, je pense, l'émergence d'un nouveau souffle dans le champ graphique et typographique.

Ressources

Bibliographie

- [B1] KINROSS R.
Modern typography: an essay in critical history.
London : Hyphen Press, 1992.
ISBN : 9780907259053.
- [B2] KAROW P, ZAPF H.
Digital typefaces: description and formats.
2nd ed. New York : Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 1994.
ISBN : 9783540565093.
- [B3] KAROW P.
Digital Formats for Typefaces. Hamburg, West Germany : URW Verlag, 1987.
ISBN : 9783926515018.
- [B4] BRINGHURST R, CLARKE J.-M, NEVEU P.
La forme solide du langage. Paris : Ypsilon, 2011.
ISBN : 9782356540164.
- [B5] MORRIS R. A, ANDRE J, ANDRI J, EDS.
Raster Imaging and Digital Typography II: Proceedings of the Conference on Raster Imaging and Digital Typography, Boston 1991 (Cambridge Series on Electronic Publishing).
1st ed. Cambridge : Cambridge University Press, 1991.
ISBN : 9780521417648.
- [B6] KINROSS R, SZIDON A.
La typographie moderne : un essai d'histoire critique. Paris : B42, 2012.
ISBN : 9782917855270.
- [B7] BILAK P, RAPPO F, KELLER P, BALDESSIN G, LUETHL P, KESHAVJEE D, TAVELLI J, SPIEKERMANN E, LEHNI J, BRUNI D.
Typeface as program: applied research and development in typography: Le caractère typographique comme programme: recherche appliquée et développement en typographie. Lausanne, Switzerland : ECAL/University of art and design, 2009.
ISBN : 9783037640722.
- [B8] ANDRÉ J.
Création de fontes en typographie numérique. Université Rennes 1, Institut de Formationsupérieure en Informatique et Communication : IRISA + IFSIC, 1993.
- [B9] ADOBE DOCUMENTATION.
DESIGNING MULTIPLE MASTER TYPEFACES. USA : Adobe, 1997.
- [B10] BIGELOW C, SEYBOLD J.
'PRINCIPLES OF DIGITAL TYPE.'
The Seybold Report. 8 Fév. 1982.

- [B11] BIGELOW C, SEYBOLD J.
'TECHNOLOGY AND THE AESTHETIC OF TYPE.'
The Seybold Report. 8 Fév. 1982
- [B12] FRUTIGER A.
Adrian Frutiger - Caracteres : L'Œuvre complete.
Basel : Birkhauser Verlag AG, 2008.
ISBN : 9783764385828.
- [B13] HARALAMBOUS Y, HORNE S. P.
Fonts and Encodings.
United States : O'Reilly Media, USA, 2007.
ISBN : 9780596102425.
- [B14] WARDE B.
'The Crystal Goblet, or Printing Should Be Invisible.' [s.l.] : British & Colonial Printer & Stationer, 1930.
- [B15] SMEIJERS F, KINROSS R,
Counterpunch: making type in the sixteenth century, designing typefaces now.
1st ed. London : Hyphen Press, 1996.
ISBN : 9780907259060.
- [B16] PROUST J.
L'Encyclopédie Diderot et d'Alembert : Planches et commentaires.
Paris : EDDL, 2001.
ISBN : 9 782 237 005 119.
- [B17] ADOBE INC.
Adobe : Type, font, format. United Kingdom : Addison Wesley, 1997.
ISBN : 9780201608540.
- [B18] KNUTH D. E.
Computers & typesetting, volume E: Computer modern typefaces.
Reading, MA : Addison-Wesley Pub. Co, 1986.
ISBN : 9 780 201 134 469.
- [B19] ADOBE INC, KADÉ A.
PostScript par l'exemple : Outils pour l'édition électronique. Paris : InterEditions, 1987. ISBN : 9 782 729 601 959.
- [B20] WOOLMAN M, BELLANTONI J.
MOVING TYPE: DESIGNING FOR TIME AND SPACE.
Switzerland : RotoVision, 2000.
ISBN : 9782880463694.
- [B21] KNUTH D. E, UNIVERSITY S.
'A Punk Meta-Font.'
TUGboat. 1988. Vol. 9, n° 2, p. 152–168.
- [B22] MARSHALL A.
Du plomb à la lumière : La Lumitype-Photon et la naissance des industries graphiques modernes.
Paris : Éd. de la Maison des Sciences de l'Homme, 2003.
ISBN : 9 782 735 110 094.
- [B23] SAVOIE A.
International Cross-currents in Typeface Design: France, Britain and the USA in the Phototypesetting Era. [s.l.] : University of Reading, 2014.
- [B24] NOORDZIJ G.
Le trait : une théorie de l'écriture.
Ypsilon éditeur, 2010.
ISBN : 9789070002176

- [B25] GERSTNER K, GREDINGER P.
*Designing programmes:
Instead of solutions for pro-
blems programmes for solution.*
3rd ed.Switzerland :
Lars Müller Publishers, 2007.
ISBN : 9 783 037 780 930.
- [B26] FEBVRE L, MARTIN H.J
L 'apparition du livre.
Paris, Albin Michel, 1958.
Bibliothèque de synthèse his-
torique. L'Évolution de l'hu-
manité, n° 49.
ISBN : 9782226047243
- [B27] POHLEN J, WELSING S., OTTO S,
ROSSI M.
*La fontaine aux lettres sur les
caractères d'impression.*
Köln : Taschen, 2011.
ISBN : 9783836525114.

Webographie

- [W1] FROSHAUG A.
'Typography is a grid | journal.'
Hyphen Press, 2000.
Disponible sur :
< [hyphenpress.co.uk/journal/
article/typography_is_a_grid](http://hyphenpress.co.uk/journal/article/typography_is_a_grid) >
(consulté le 19 Fév. 2016)
- [W2] AHRENS T.
« Une méthode basée sur les
polices *Multiple Master* pour
mettre à l'échelle des glyphes
sans en changer les caractéris-
tiques de trait. »
Document numérique. 1
April 2006. Vol. 9, n° 3-4, p. 133-
165.
Disponible sur :
< [dx.doi.org/10.3166/
dn.9.3-4.133-165](http://dx.doi.org/10.3166/
dn.9.3-4.133-165) >
- [W3] BLANCHARD G.
« *Similar to...* » ou l'euro-typo-
graphie à l'ère de la « *Postscript
Life.* »
Communication et langages.
1991. Vol. 89, n° 1,
p. 82-93. Disponible sur :
< [dx.doi.org/10.3406/
colan.1991.2319](http://dx.doi.org/10.3406/
colan.1991.2319) >
- [W4] RODRÍGUEZ VALERO D.
« *Systèmes typographiques
intelligents et souples.* »
Document numérique
1 April 2006. Vol. 9, n° 3-4,
p. 117-132.
Disponible sur :
< [dx.doi.org/10.3166/
dn.9.3-4.117-132](http://dx.doi.org/10.3166/
dn.9.3-4.117-132) >
- [W5] ANDRÉ J, DELORME C.
« *Le Delorme, un caractère
modulaire et dépendant du
contexte.* »
Communication et langages.
1990. Vol. 86, n° 1, p. 64-76.
Disponible sur :
< [dx.doi.org/10.3406/
colan.1990.2261](http://dx.doi.org/10.3406/
colan.1990.2261) >
- [W6] FRUTIGER A.
« *La forme des caractères à
l'âge de la photocomposition.* »
Communication et langages.
1969. Vol. 4, n° 1, p. 45-53.
Disponible sur :
< [dx.doi.org/10.3406/
colan.1969.3766](http://dx.doi.org/10.3406/
colan.1969.3766) >
- [W7] ANDRÉ J.
« *Nouveaux caractères pour
une nouvelle typographie.* »
Linx. 1991. Vol. 4, n° 1, p. 45-54.
Disponible sur :
< [dx.doi.org/10.3406/
linx.1991.1187](http://dx.doi.org/10.3406/
linx.1991.1187) >
- [W8] BELLA G, HARALAMBOUS Y.
« *Fontes intelligentes, textèmes
et typographie dynamique.* »
Document numérique
. 1 April 2006. Vol. 9, n° 3-4,
p. 167-216.
Disponible sur :
< [dx.doi.org/10.3166/
dn.9.3-4.167-216](http://dx.doi.org/10.3166/
dn.9.3-4.167-216) >

- [W9] LARSON K, PICARD R. W.
'*The Aesthetics of Reading.*'
In : People and Computers
XX — Engage . [s.l.] : Springer
Science + Business Media,
2007. p. 41–56. Disponible sur :
< dx.doi.org/10.1007/
978-1-84628-664-3_4 >
ISBN : 9781846285882.
- [W10] DARDAILLER D.
« *Normes et fontes.* »
Cahiers GUTenberg
1989. n°4, p. 2–8. Disponible
sur :
< dx.doi.org/10.5802/cg.41 >
- [W11] COLES S, STOWELL S, DANIELS S,
HALLMUNDUR A, WEISSMAN D,
BARBER P, PAPAELIAS A, ULRICH
F, RAPOSO T, HOFWEBER P,
RENDLE R, LEINSTER T.
'*Beyond Helvetica: The Real
Story Behind Fonts in iOS 7.*'
Typographica, 2013.
Disponible sur :
< typographica.org/on-typo-
graphy/beyond-helvetica-
the-real-story-behind-fonts-
in-ios-7/ >
(consulté le 15 Sept. 2015)
- [W12] RIGGS T. 'CELEBRATING 25
Years of Adobe Originals.'
The Typekit Blog, 2014.
Disponible sur :
< blog.typekit.com/25-years-of-
adobe-originals/ >
(consulté le 15 Sept. 2015)
- [W13] FÁBIO.
'*Bezier Curves and Type
Design: A Tutorial | Learn –
Scannerlicker!*'
Disponible sur :
< learn.scannerlicker.
net/2014/04/16/bezier-
curves-and-type-design-a-tu-
torial/ >
(consulté le 15 Sept. 2015)
- [W14] DUARTE MARTINS F.
'*Bezier Curves and Type
Design: A Tutorial | Learn –
Scannerlicker!*'
Disponible sur :
< learn.scannerlicker.
net/2014/04/16/bezier-
curves-and-type-design-a-tu-
torial/ >
(consulté le 15 Sept. 2015)
- [W15] ORMANDY K.
« *Efficient Web Type, c. 1556.* »
Kenneth Ormandy, 2015.
Disponible sur :
< kennethormandy.com/
journal/efficient-web-type-
circa-1556 > (consulté le 15
Sept. 2015)
- [W16] SMITH R, HUDSON J, CROSSLAND
D, TWARDUCH A, BERLOW
D, INDRAKUPFERSCHMID,
LOZOS C, PUCKETT J, TÉTAR A,
SÖDERSTRÖM G.
License violation.
May 2015.
Disponible sur :
< typedrawers.com/dis-
cussion/comment/12339 >
(consulté le 12 Janv. 2016)

- [W17] GUESNON B, TWARDOCH A, BERLOW D, HOSNY K, HUDSON J, PHINNEY T, CROSSLAND D, JOHNSON A, SEIFERT G, LOUISREMI, SIMONSON M, PUCKETT J.
Looking for some info on MM fonts and AAT format
November 2015. Disponible sur : < typedrawers.com/discussion/comment/16361 > (consulté le 12 Janv. 2016)
- [W18] CROSSLAND D, JENNINGS J, TÉTAR A, FISTER L, ROLEK G, TWARDOCH A, VOKITS M, LARABIE R, HUDSON J, SHAVIT O, THOMAS G, FINK R, BERLOW D.
Introducing Trufont, World's First native UFO3 font editor (& fully cross-platform)
. October 2015.
Disponible sur : < typedrawers.com/discussion/comment/15650/ > (consulté le 12 Janv. 2016)
- [W19] NATHAN W.
'Debating OpenType 2.0'
[s.l.] : [s.n.], 2015.
Disponible sur : < lwn.net/Articles/662813/ > (consulté le 12 Janv. 2016)
- [W20] SHERMAN N, MORE.
'Font hinting and the future of responsive typography.'
[s.l.] : Google+, 2013.
Disponible sur : alistapart.com/column/font-hinting-and-the-future-of-responsive-typography > (consulté le 12 Janv. 2016)
- [W21] SHERMAN N.
'Variable fonts for responsive design.' Google+, 2015.
Disponible sur : < alistapart.com/blog/post/variable-fonts-for-responsive-design > (consulté le 12 Janv. 2016)
- [W22] SHERMAN N.
'The future of responsive typography.'
Disponible sur : < nicksherman.com/AEA14/ > (consulté le 12 Janv. 2016)
- [W23] ENG T.
'InDesign, the hz-program and Gutenberg's secret.'
Typography i Norge.
20 Sept. 2009.
Disponible sur : < www.typografi.org/justering/gut_hz/gutenberg_hz_english.html > (consulté le 25 Janv. 2016)
- [W24] VAN BLOKLAND E, VAN ROSSUM J.
'Is Best Really Better.', 1990.
Disponible sur : < lettererror.com/writing/is-best-really-better/ > (consulté le 19 Fév. 2016)
- [W25] IA.
'Web design is 95% typography.', 2006.
Disponible sur : < ia.net/know-how/the-web-is-all-about-typography-period > (consulté le 19 Fév. 2016)

- [W26] SHERMAN N, LEWIS C.
'Font-to-width.', 2013.
Disponible sur :
< font-to-width.com/ >
(consulté le 19 Fév. 2016)
- [W27] SHERMAN N.
'Size calculator.', 2013.
Disponible sur :
< sizecalc.com/ >
(consulté le 19 Fév. 2016)
- [W28] SHERMAN N.
'The future of responsive typography.', 2014.
Disponible sur : < nicksherman.com/AEA14/ >
(consulté le 19 Fév. 2016)
- [W29] SHERMAN N.
'Font hinting and the future of responsive typography.'
Google+, 2013.
Disponible sur :
< alistapart.com/column/font-hinting-and-the-future-of-responsive-typography >
(consulté le 19 Fév. 2016)
- [W30] SHERMAN N.
'Responsive typography is a physical discipline, but your computer DƐsn't know it (yet).'
- [W31] STEIN B.
'State of web type.', 2014.
Disponible sur :
< www.stateofwebtype.com/ >
(consulté le 19 Fév. 2016)
- [W32] WILLIS N.
'Debating OpenType 2.0', 2015.
Disponible sur :
< lwn.net/Articles/662813/ >
(consulté le 19 Fév. 2016)
- [W33] VAN ROSSUM J, VAN BLOKLAND E.
« MutatorMath. » LettError, 2014.
Disponible sur :
< lettererror.com/2014/09/19/mutatormath/ >
(consulté le 19 Fév. 2016)
- [W34] SOUCHIER E.
« L'image du texte pour une théorie de l'énonciation éditoriale. »
Les cahiers de médiologie. 1998. Vol. 6, n° 2, p. 137.
Disponible sur :
< dx.doi.org/10.3917/cdm.006.0137 >
- [W35] ANDRÉ J, OSTROMOUKHOV V.
« Punk : De METAFONT à postScript. »
Cahiers GUTenberg. 1989. n°4, p. 23–28.
Disponible sur :
< dx.doi.org/10.5802/cg.43 >
- [W36] LETTEROR.
'LettError: Responsive lettering, Typographics 2015.'
[s.l.] : [s.n.], 2015.
Disponible sur :
< lettererror.com/dev/mathshapes/ >
(consulté le 19 Fév. 2016)
- [W37] BERLAEN F.
« RoboDocs — RoboDocs. »
[s.l.] : [s.n.], 2015.
Disponible sur :
< www.robodocs.info >

- [W38] LETTERROR.
'Bad type.' LettError, 2015.
 Disponible sur :
 < letterror.com/writing/bad-type >
 (consulté le 19 Fév. 2016)
- [W39] MIDDENDORP J.
'Tools.'
 In : NYPELS C, ED.
 LettError, 2012.
 Disponible sur :
 < letterror.com/writing/toolspace/ >
 (consulté le 19 Fév. 2016)
- [W40] WILD A.
 « *La typographie de la bible de Gutenberg.* »
 Communication et langages.
 1997. Vol. 114, n° 1, p. 85–95.
 Disponible sur :
 < dx.doi.org/10.3406/colan.1997.2810 >
- [W41] PHINNEY T.
'Most overlooked typefaces: Penumbra.'
 The Typekit Blog, 2005.
 Disponible sur :
 < blog.typekit.com/2005/11/22/most-overlooked/ > (consulté le 25 Fév. 2016)
 Jan Middendorp
 LettError Book. Charles Nypels, 2000. Disponible sur :
 < <http://letterror.com/writing/toolspace/> >
 (consulté le 25 Fév. 2016)
- [W42] BILAK P.
'Typotheque: The history of history by Peter Bílak.'
 In : Typotheque , 2010.
 Disponible sur :
 < typotheque.com/articles/the_history_of_history >
 (consulté le 25 Fév. 2016)
- [W43] BOARDLEY J.
'Better UI for better typography — I love typography.'
 In : typography . I Love Typography, 2014.
 Disponible sur :
 < <http://ilovetypography.com/2014/10/22/better-ui-for-better-typography-adobe-petition/> >
 (consulté le 25 Fév. 2016)
- [W44] CROSSLAND D, JENNINGS J, TÉTAR A, FISTER L, ROLEK G, TWARDOCH A, VOKITS M, LARABIE R, HUDSON J, SHAVIT O, THOMAS G, FINK R, BERLOW D.
'Introducing Trufont, World's First native UFO3 font editor (& fully cross-platform).'
 In : Type Drawers .
 TypeDrawers, 2015.
 Disponible sur :
 < typedrawers.com/discussion/comment/15650 >
 (consulté le 25 Fév. 2016)
- [W45] LAURENCE PENNEY
'OpenType.'
 In : Truetype Typography [En ligne], 1997.
 Disponible sur :
 < <http://www.true-type-typography.com/opentype.htm> >
 (consulté le 28 February 2016)

- [W46] HUDSON J, BERLOW D, SHINN N, PABLOIMPALLARI, SEIFERT G, PENNEY L, KOSOFSKY S.-M, THOMAS G, CEROVICH S, USER D, CROSSLAND D.
'*New Microsoft size-specific design selection mechanism.*'
TypeDrawers, 2013.
Disponible sur :
< typedrawers.com/discussion/470/new-microsoft-size-specific-design-selection-mechanism/p1 >
(consulté le 25 Fév. 2016)
- [W47] MICROSOFT.
'*Microsoft typography — what is TrueType?*'
7 July 2009.
Disponible sur : < www.microsoft.com/en-us/Typography/WhatIsTrueType.aspx > (consulté le 28 Fév. 2016)
- [W48] MICROSOFT.
'*Microsoft typography - A brief history of TrueType.*'
30 June 1997.
Disponible sur : < www.microsoft.com/typography/TrueTypeHistory.mspx >
(consulté le 28 Fév. 2016)
- [W49] MACMILLAN D. M., KRANDALL R.
'*The Benton Pantographs.*' In : *Circuitous Root*
[En ligne] Disponible sur :
< www.circuitousroot.com/artifice/letters/pantocut/benton/index.html >
(consulté le 28 Fév. 2016)
- [W50] MACMILLAN D. M., KRANDALL R.
'*The Benton Pantographs.*' In : *Circuitous Root*
[En ligne] Disponible sur :
< www/artifice/letters/press/typemaking/mats/optical/index.html >
(consulté le 28 Fév. 2016)
- [W51] HUDSON J.
'*New Microsoft size-specific design selection mechanism.*'
TypeDrawers, 2013.
Disponible sur :
< typedrawers.com/discussion/470/new-microsoft-size-specific-design-selection-mechanism/p1 >
(consulté le 1 Mars 2016)
- [W52] P. BILAK
'*The History of history.*'
Typothèque, 2010.
Disponible sur :
< www.typotheque.com/articles/the_history_of_history >
(consulté le 1 Mars 2016)

Videographie

- [v1] VAN BLOKLAND E.
Optics and code .
2015. Disponible sur :
< vimeo.com/107745440 >
(consulté le 19 Fév. 2016)
- [v2] CLIMER A.
*Designing obsidian with
Andy Clymer* .
24 March 2015.
Disponible sur :
< vimeo.com/124062807 >
(consulté le 19 Fév. 2016)
- [v3] WILSON D.
*The problem with hand com-
position* .
3 July 2012.
Disponible sur :
< vimeo.com/43461980 >
(consulté le 19 Fév. 2016)
- [v4] COMPANY A. T. F, GRAUER B.
Type speaks . 1948.
Disponible sur :
< vimeo.com/133085076 >
(consulté le 19 Fév. 2016)
- [v5] OUTOFSORTSFILM.
*Punchcutting at the atelier
press & Letterfoundry*
. YouTube.
13 Sept. 2009.
Disponible sur :
< [www.youtube.com/
watch?v=eExllUeGtvc](https://www.youtube.com/watch?v=eExllUeGtvc) >
(consulté le 19 Fév. 2016)
- [v6] INTERNATIONAL TYPOGRAPHICAL
UNION.
I.T.U. - story of technology
. 1966.
Disponible sur :
< vimeo.com/128146910 >
(consulté le 19 Fév. 2016)
- [v7] FONTLAB LTD.
*TrueType GX variations
Jam by Erik van Blokland*
. 2015.
Disponible sur :
< vimeo.com/120047889 >
(consulté le 19 Fév. 2016)
- [v8] VAN BLOKLAND P.
*Robothon 15 01.02 keynote:
Extrapolation* . 2015.
Disponible sur :
< vimeo.com/123816124 >
(consulté le 19 Fév. 2016)
- [v9] VAN BLOKLAND E.
*Robothon 15 01.05
MutatorMath Erik van Blokland*
. 2015.
Disponible sur : < [vimeo.
com/123815114](https://vimeo.com/123815114) > (consulté le
19 Fév. 2016)
- [v10] SHERMAN N.
*Robothon 15 02.05 variable
fonts for responsive design*
. 2015.
Disponible sur :
< vimeo.com/123813231 >
(consulté le 19 Fév. 2016)

- [V11] LEMING T, VAN BLOKLAND E.
*Robothon09 Prepolator
Superpolator* .
5 March 2015.
Disponible sur :
< vimeo.com/116066509 >
(consulté le 19 Fév. 2016)
- [V12] FONTLAB LTD.
*Ares FontChameleon 1.5 on
windows 98* . 2010.
Disponible sur :
< vimeo.com/14758983 >
(consulté le 19 Fév. 2016)
- [V13] FONTLAB LTD.
*TrueType GX variations
Skia by Matthew Carter*
. 2015.
Disponible sur :
< vimeo.com/120047887 >
(consulté le 19 Fév. 2016)
- [V14] KNEBUSCH J.
Instant Type Family
. 2013.
Disponible sur :
< vimeo.com/89250074 >
(consulté le 19 Fév. 2016)
- [V15] PHINNEY T.
*Why OpenType Succeeded
Where GX & MM Didn't*
. 1 November 2015.
Disponible sur :
< vimeo.com/141448515 >
(consulté le 19 Fév. 2016)

Citations

- [B12] FRUTIGER A.
Adrian Frutiger - Caractères.: L'Œuvre complete
- [C1] *Quand il créa un caractère pour photocomposition sur monotype ou linotype, le dessinateur est limité par le système à 18 unités dans lequel tout signe s'inscrit, tandis que sur lumitype, il a 35 unités à sa disposition — c'est à dire le double. Les lettres à crénage ne posent pas de problème non plus. Les particularités de la technique photographique doivent toutefois être prises en compte lors de la réalisation du dessin original lui même, dont la forme doit être adaptée : il faut élargir les encoches trop fines, qui ont tendance à se boucher au flashage. À l'inverse, les angles droits arrondissent si l'éclair est insuffisant. Le dessinateur de caractères peut y remédier en rajoutant des pointes. Autre problème : les très fins déliés qui n'ont pas la densité requise au flashage et menacent de se rompre à la copie sur plaque doivent être dessinés plus épais.*
- [C2] En revanche, il n'y a pas eu de lettres en bois pour l'ondine, alors que nous en avons chez deberry et peignot, même si elles n'étaient pas fabriquées sur place
- [W1] FROSHAUG A.
« *Typography is a grid* | *journal.* » 2000.
- [C3] *To mention both typographic, and, in the same breath/sentence, grids, is strictly tautologous. The word typography means to write/print using standard elements; to use standard elements implies some modular relationship between such elements; since such relationship is two-dimensional, it implies the determination of dimensions which are both horizontal and vertical. Consider the problems which faced Gutenberg, some five hundred years ago, in helping 'the eternal God' to bring 'into existence the laudable art, by which men now print books, and multiply them so greatly ...'.
1 — Item, the said Johann Gutenberg knew of the invention of paper (which had reached Cologne by 1320); item, knew of the development of suitable inks ... of the general features of the cloth- and wine-press, of the*

arts of the engravers, of the die- and punch-making of the goldsmiths (after all, he was a goldsmith himself).

2 — What did Gutenberg invent? In order that letters, characters, may be arranged in lines, line upon line, for printing, each letter must be of the same depth or body-size as its neighbours, irrespective of its individual width: the vertical dimension (y in Cartesian co-ordinates) is critical. If, as seems historically probable, Gutenberg's invention was that of the adjustable type-mould, tolerant of characters of differing widths, intolerant of divergence in body-size.

3 — This invention acted as a vertical grid upon the setting, the forme, the page. But the length of line, the width of setting, provided another dimension. It seems that this horizontal dimension (x) of the grid was determined by convention, and embodied in the Procrustean bed of the composing stick – probably at that time, as more lately in the case of 13-pica fixed newspaper sticks, an unadjustable hod into which the standard bricks of characters could be successively piled.

[W38] JAN MIDDENDORP
LettError Book

[C4] *Software is a commercial mass product, so it is not made for individuals but for a target group with hypothetical wishes and demands. If*

in practice that target group turns out to have unfulfilled desires extra functions are added with an upgrade, but only if there is enough interest. (...) In the past you were successful if you could make what was in your mind's eye with your own hands. Now there is a danger that designers will only function properly if their creativity does not exceed the potential of their computer: if the world of ideas fits inside the world of tools. Of course, in the long run this leads to a monotonous computer-driven uniformity.

[W22] ERIK VAN BLOKLAND
(2013) ATypI conference,
<twitter.com/NickSherman/status/389651088401764352>

[C5] *If you design a single font, it's an island. If you design more than one, you're designing the relationship, the recipe.*

[C6] *Technically we can create the slickest printing ever, reach the highest possible quality ever. Unfortunately, the results have too often become absolutely boring. The quality of a printed product, the high resolution of its typeface, the perfect printing are not necessarily what makes for good design or clear communication. »*

[W24] VAN BLOKLAND E, VAN ROSSUM J.
'Is Best Really Better.', 1990.

[C5] Through our experience with traditional typesetting method,

we have come to expect that the individual letterforms of a particular typeface should always look the same. This notion is the result of a technical process, not the other way round. However, there is no technical reason for making a digital letter the same every time it is printed. It is possible to calculate every point and every curve differently each time the letter is generated by slightly moving the points that define a character in various “random” directions.

[c6] *A design usually starts with a series of sketches, a vague visualization of possibilities that gets the selection process — the choice between good and mediocre solutions — going. As long as you're sketching on a napki, the marks can mean whatever you like. If you start straight away with the mouse in your hand, which is more common nowadays, you are immediately confronted with tenths of millimeters and dead straight lines. The design already takes on finality when it is time for flexibility. [...] There is an alternative: programming-assisted design. In this case the alternatives are neither sketched by hand nor directly drawn on the computer. The designer indicates the ingredients and the margin, and asks the computer to think up a series of variables, taking certain rules into account. In other words, to vary certain parameters at random.*

The computer itself does not start designing, it cannot be accused of interfering with the creative process. It does what a computer is best at doing: very quickly working out the consequences of possible decisions.

[W17] GUESNON B, PHINNEY T.
Looking for some info on MM fonts and AAT format

[c7] [Baptiste Guesnon]
Hello,
It seems that these formats are more or less dying (or already dead). I don't understand if the MMFont of Adobe is still supported in the OpenType format, or if only works with Type 1. Does someone know about it?

[Thomas Phiney]
Adobe's multiple master format was supported in some early versions of OpenType, but that was withdrawn very early on. I don't think there is much of anything that supports it.

Is it possible to build an MM Master AAT with two UFO master, or two TTF for example?

And, if it is possible, is there a way to use it? Or should I build an app from myself with Cocoa and Core text to do it?

Multiple Master specifically refers to the Adobe tech. "Multiple Master AAT" is pretty much a contradiction

in terms. I am guessing that what you really want is an AAT Variations font, which is the AAT equivalent of MM. So, currently, the only tools I know that support building such fonts are command-line tools from Apple, that take TrueType inputs. However, you could easily create such from UFO sources, one step upstream. As for testing, I think you'll find that the fonts will work in many Mac apps that use CoreText APIs, including Apple's own Pages, etc.

I don't really understand why MM Fonts (PST1 seems to be dead so AAT) are not more used. The idea of MM has an amazing potential, but for now it failed to rise up each time. The lack of tools to produce and use it seems the major obstacle to me. Maybe someone has some information about it?

There most definitely were adequate tools to build MM fonts. FontLab Studio supported MM fonts reasonably well, and most of the commercial fonts available in the world have been produced with FontLab Studio in part or in whole.

The problems for MM historically were failures of user interest, marketing, and app support for direct axis access. Adobe didn't realize how hard it would be to make MM fly, and didn't take the right steps to make it successful. (This

harsh, sad lesson would later serve Adobe well in promoting OpenType.)

AAT (GX at the time) is a slightly different story, but it did have the tools problem you mention, plus it was originally a Mac-only technology when top apps were all cross-platform, so Adobe, Quark and Microsoft all failed to get behind it.

Interestingly, axis-based fonts might actually make a comeback. Some of the talk about OpenType 2.0 leans in this direction, and support for the AAT gvar table has been increasing recently (I think we now have it in TTX, HarfBuzz and the Universal Shaping Engine).

[B22] FEBVRE L, MARTIN H.J
L'apparition du livre.
Page 109

[C8] [...] *les signes typographiques étaient beaucoup plus nombreux qu'aujourd'hui - car le désir d'imiter les écritures manuscrites incitait les typographes à faire fondre ensemble des lettres réunies par des ligatures et l'emploi des abréviations (ã = an ou am; q = quia, etc.), avait des conséquences analogues. Et l'on peut se demander si l'abandon progressif de l'emploi de ces ligatures et des ces abréviations, si nombreuses dans les livres imprimés du XVe siècle et du début du XVIe siècle, n'eut pas en partie pour origine le désir de diminuer le nombre*

- des poinçons à tailler et des matrices à frapper : manifestation de cette tendance vers l'uniformisation et la simplification qui caractérise, dans bien des domaines, l'évolution du livre et de son industrie.*
- [B18] MARSHALL A.
Du plomb à la lumière : La Lumitype-Photon et la naissance des industries graphiques modernes.
Page 8-9, Introduction.
- [C9] *Quand à l'idée selon laquelle la technique « optimale » va toujours finir par s'imposer en vertu d'une sorte de processus de sélection naturelle, l'histoire des techniques ne manque pas de contre-exemples démontrant que ce n'est pas forcément la technique la plus performante qui connaît le plus grand succès sur le plan économique. [...] Force est de reconnaître que l'innovation n'est pas un processus linéaire gouverné par une logique implacable d'optimisation technique. Certes, la tendance est généralement vers une plus grande efficacité (notion qui doit être comprise dans le sens économique, mais aussi technique). Mais sa progression est souvent plutôt arborescente, son chemin étant jalonné de choix qui s'ouvrent aux ingénieurs et aux entrepreneurs au gré des opportunités technique supposée optimale.*
- [B18] Page 13, Introduction.
- [C10] *Gutenberg vit à une époque où la grande nouveauté est la fabrication de produits en série — les caractères mobiles sont de ceux-là. Son invention est issue d'une idée simple : la multiplication de caractères répondants à des normes identiques. [...] La facilité de création de nouveaux caractères ne va-t-elle pas diminuer finalement les qualités artistiques en les diversifiant ? Ce problème, ancien, qui valait déjà suscitait au milieu du XIXe siècle, au moment de la mécanisation de la presse, l'indignation de Charles Nodier, est plus que jamais d'actualité de nos jours où se banalisent les outils numériques de création de caractères, d'autant que l'apparition de l'écran, le retour de l'oral par le biais du multimédia, la communication immédiate à distance engendrent une révolution aussi essentiel que le remplacement du rouleau de papyrus, le volume, par le cahier de parchemin.*
- [wx] http://issuu.com/lettererror/docs/lettererror_autobrander_2001
- [C11] *Anything that can not be automated is design*
Petr van Blokland
Anything that can be automated, will be automated
Letterror

[c5] *Ce n'est pas toujours le cas, mais le plus souvent du matériel est confié au designer graphique, sur lequel il travaille et opère une sélection. Le design n'est habituellement pas une création partant de zéro. [...] Autre chose doit être ajouté au sujet du langage, car l'idée clé, le mot-clé de cette conférence — éditer — se retrouve différemment au sein même des langues. En anglais, sans doute, l'éditeur est celui qui traite le matériel édité, le trie, l'agence, travaille à en réduire la taille ou demande à l'étendre, enfin y met bon ordre. En français il me semble que sous ce terme sont superposées deux notions : celle qui vient d'être évoquée et le fait de publier. En français, un « éditeur » est ce que nous nommons en anglais un « publisher » et je peux supposer que cette signification témoigne d'une persistance d'une idée historique de la publication. C'est l'idée que l'« éditeur » tel qu'il travaillait avant l'avènement des grandes maisons d'édition — tant comme éditeur des textes à proprement parler qu'en tant que celui qui prend soin de la production des objets, mais encore de leur distribution et leur vente. Chez moi ces deux fonctions se combinent, bien que cela devienne moins fréquent et que je ne tienne pas*

Biographies

- [BIO 1] Gerrit Noordzij est un designer typographique néerlandais ayant enseigné à l'Académie royale des beaux-arts de La Haye aux Pays-Bas de 1960 à 1990 en tant que directeur du *Programme d'écriture et de calligraphie*. Il est l'auteur d'un livre théorisant le lien entre l'écriture et le design typographique avec le trait. Il présenta cette théorie dans l'édition *The stroke of the pen : fundamental aspects of western writing en 1982* et *De Streek: Theorie van het schrift* en 1985. Il fut notamment le professeur d'Erik Van Blokland et Just Van Rossum, le duo formant Letterror.
- [BIO 2] Pierre Bézier est un ingénieur français inventeur des courbes de Bézier en 1962 dans le cadre de son travail chez le constructeur automobile Renault.
- [BIO 3] Paul de Faget de Casteljau est un mathématicien et physicien français. Il est connu pour sa découverte de l'algorithme de Casteljau qui permet de représenter les courbes de Bézier dès 1958 alors qu'il travaillait pour Citroën.
- [BIO 4] Jay David Bolter est un auteur Américain travaillant sur l'évolution des médias, l'utilisation des technologies dans l'éducation et le rôle de l'outil informatique dans le processus de l'écriture. Il est notamment l'auteur de *Turing's Man : Western Culture in the Computer Age* (1984), *Writing Space: The Computer, Hypertext, et History of Writing* (1991).
- [BIO 5] Erik Van Blokland est un des deux membres de Letterror.
- [BIO 6] Just Van Rossum est un des deux membres de Letterror. Il est également le frère de Guido Van Rossum, le créateur du langage Python.
- [BIO 7] Letterror est un duo de designers typographiques néerlandais à l'origine de nombreuses expérimentations et outils informatiques sur la typographie. Leur approche du design par la programmation les a mené à réassembler de nombreux projets concernant la typographie dynamique. On peut citer *le Beowolf*, *Superpolator*, *Robofab*, *Digital Lettering* ou encore la plateforme de live-coding python *Drawbot*.

- [BIO 8] Jan Middendorp est un écrivain, éditeur et consultant néerlandais indépendant. Il est l'auteur d'articles et de livres tels que *Dutch Type* (2004), *Shaping Text* (2012) ou encore la série d'articles *Creative Characters* (2010) pour myFonts.
- [BIO 9] Donald Knutt est un informaticien et mathématicien Américain professeur à la faculté de l'université de Stanford. Il est l'un des pionniers de l'algorithmique et est l'auteur des logiciels libres TeX et Metafont.
- [BIO 10] Jacques André est un informaticien et mathématicien français chercheur à l'INRIA de Rennes. Il s'intéressa à la typographie et aux questions du potentiel dynamique de la typographie digitale. Il réalisa de nombreuses expérimentations à ce sujet à travers petits programmes informatiques durant les années 80 et 90 et publia de nombreux textes à ce sujet. Il est également l'auteur d'ouvrages sur l'histoire de la typographie.
- [BIO 11] Richard Southall est un designer typographique qui prit part, au cours de sa carrière, aux nombreuses révolutions technologiques typographiques qui traversèrent le 20e siècle. Il travailla sur la Lumitype photo, le projet Metafont avec Donald Knutt et au Xerox Park de Palo Alto. Il fut également enseignant du département de design typographique à l'université de Reading et auteur d'un livre de bibliorence sur l'histoire des évolutions techniques de la typographie au cours du 20e siècle : *Printer's Type in the Twentieth Century: Manufacturing And Design Methods*.
- [BIO 12] Robert Bringhurst est un typographe, poète, éditeur et linguiste canadien. Il est l'auteur de *The Elements of Typographic Style* (1992) et de *The Solid Form Of Language : An Essay On Writing And Meaning* (2004).
- [BIO 13] Gerard Unger est un typographe, designer typographique et auteur néerlandais. Il est l'auteur des fontes de texte *Swift*, *Gulliver* ou encore *ANWB-fonts* pour les autoroutes néerlandaises.
- [BIO 14] Fred Smeijers est un designer typographique et chercheur néerlandais. Il est notamment l'auteur de la famille de caractères *Quadrat*, *Arnhem* et *Nobel*. Il a mené de nombreuses recherches sur les contrepointons et est l'auteur du livre *Counterpunch, making type in the sixteenth century, designing typefaces now* (1996) où il expose son travail de recherche et l'influence qu'a eu ce dernier sur son travail de designer typographique.

- [BIO 15] Adrian Frutiger est un typographe suisse né en 1928 et mort en 2015. Il commença sa carrière à la fonderie Deberny & Peignot à Paris où il réalisa la conversion et la création et de nombreux dessins du caractère au plomb pour la photocomposition. Il est également l'auteur de la famille de caractère *Univers* (1957), première « super famille » pensée dans toutes ses déclinaisons de graisses et de largeur au moment de sa conception.
- [BIO 16] Matthew Carter est un designer typographique anglais vivant et travaillant aux États-Unis. Il est l'auteur de nombreuses fontes de renoms tels que le *Verdana*, le *Tahoma*, le *Georgia* ou le *Big Caslon*. Au cours de sa carrière, il vit la fin de l'ère au plomb, la photocomposition et la typographie digitale.
- [BIO 17] Hermann Zapf est un designer typographique allemand né en 1918 et mort en 2015. Il est l'auteur de nombreuses typographies telles que l'*Aldus*, l'*Optima*, le *Palatino* ou le *Zapfino*. Au cours de sa carrière, il vécut la transition du plomb à la photocomposition puis au numérique. Il participa avec Peter Karrow à la création de l'algorithme de justification et de césure *Hz* (pour *Herman Zapf*).
- [BIO 18] Peter Karrow est un informaticien allemand spécialisé dans les technologies de création de fontes. Il travailla à la fonderie de caractère URW à l'origine du format Ikarus. Il est l'auteur de plusieurs ouvrages sur la technologie numérique en typographie.
- [BIO 19] Thomas Phinney est un consultant expert en typographie digitale. Il a travaillé à Adobe en tant que Chef de produit. Il a participé à l'élaboration de nombreux formats tels que le format *Multiple Master Fonte*.
- [BIO 20] Nick Sherman est un typographe et consultant typographique américain. Il a travaillé sur le design web de Font Bureau, Webtype et MyFonts et est très impliqué dans le développement de la typographie sur et pour le web.

Lexique

- [LEX1] **La typographie**
La typographie est une discipline et un outil permettant la transcription visuelle de texte. La typographie est composée de types. Elle peut également être une forme graphique en soit, un ensemble de signes.
- [LEX2] **Une typographie**
Une typographie est un ensemble de caractères formant un ensemble cohérent.
- [LEX3] **Typographe**
Un typographe utilise la typographie pour mettre en page du texte.
- [LEX4] **Designer typographique**
Le designer typographique est celui qui crée des typographies.
- [LEX5] **Fonte**
Une fonte est un fichier numérique contenant la description numérique de tous les glyphes et de leurs caractéristiques d'une typographie.
- [LEX6] **Format de fonte**
Il existe de nombreux formats de fontes, et donc de nombreuse manière de procéder à la description des glyphes et de leur interaction.
- [LEX7] **Glyphe**
Un Glyphe désigne le dessin, l'image d'un type.
- [LEX8] **Type**
Un type désigne un glyphe en tant qu'entité conceptuelle. Par exemple la lettre A, quel que soit son dessin.
- [LEX9] **Caractère typographique**
Un caractère typographique est un ensemble complet de types regroupé dans une même casse, ou un même fichier fonte.
- [LEX10] **Famille typographique**
Une famille typographique est un ensemble de caractères typographiques unis par une même logique et un même système graphique. Une famille typographique forme une unité.
- [LEX11] **Superfamille typographique**
Une super-famille est une famille typographique au très grand nombre de variantes se déclinants dans plusieurs axes (Avec et sans empattements, Graisses, Largeurs, Script, etc.)

- [LEX12] Corps
Le corps d'un caractère est la mesure de la hauteur d'un caractère. Directement issue du plomb, cette mesure est exprimée en points typographiques.
- [LEX13] Corps optique
Le corps optique d'un caractère correspond à la forme d'un glyphe à un corps donné.
- [LEX14] Mise à l'échelle
La mise à l'échelle d'une glyphe correspond à son agrandissement ou rétrécissement. Celui peut-être mis à l'échelle de manière homothétique, anamorphique ou algorithmique.
- [LEX15] Gris typographique
Le gris typographique est le gris optique obtenu par la juxtaposition des glyphes formants une trame une fois composée sous forme de texte. La densité de noir contenu dans le dessin des glyphes pourra faire varier la noirceur du gris typographique.
- [LEX16] Forme/Contreforme
La contreforme d'un glyphe correspond au blanc encerclé par le noir de son dessin (sa forme). Seuls les glyphes ayant une surface de blanc fermé ont une contreforme. Habituellement le o, b, d, r, p et q.
- [LEX17] Fût
Le fût est le trait vertical principale d'un glyphe, comme dans le L, I, h, F, etc.
- [LEX18] Montantes/Descendantes
Les montantes et descendante sont les parties des bas de casses dépassant en hauteur la hauteur d'x ou la ligne de base.
- [LEX19] Point typographique
Un point typographique est une unité de mesure. Un point typographique correspond à 0,352778 mm. Il existe également un système décimal utilisé à l'imprimerie nationale, ainsi que le système Didot (0,0249 % plus large que le point typographique).
- [LEX20] Mémoire cache
La mémoire cache est une mémoire qui enregistre temporairement des données afin de diminuer le temps d'accès et d'éviter la répétition d'un même calcul (comme celui de la forme d'un glyphe dans le cas des fontes Postscript).
- [LEX21] Les points bornes
Il existe différents type de points. Les points qui définissent le segment (*oncurve*), et les points qui définissent la courbe de Bézier (*offcurve*).
- [LEX22] Rendu de sous-pixel
Le rendu de sous-pixel est une manière d'améliorer la résolution apparente d'un écran en utilisant les canaux (Rouge, Vert et Bleu) com-

posant les pixels. L'utilisation de ces canaux en fonction de leur position sur la dalle et non plus pour leur couleur, permet de rendre les courbes d'un tracé plus fluide et plus nettes. Il est possible d'afficher des glyphes sans rendu sous pixel (affichage Bitmap), en utilisant seulement des niveau de gris ou en utilisant des technologies d'anti-aliasing.

[LEX23] Anti-aliasing

Il existe actuellement différent de moteurs d'anti-aliasing. Les principaux sont : Cleartype (Windows), Quartz (Apple), Cooltype (Adobe) et Freetype (libre). Ce sont ces environnement logiciels qui gèrent l'utilisation du rendu sous pixel et donc la manière dont seront interprétés et affichés les images et glyphes à l'écran.

Annexe

[ANN1] Pantographe

Un pantographe est un outils de dessin formé de tiges articulées permettant de réaliser des agrandissement ou des réductions homothétiques d'un dessin. Le premier pantographe a été construit en 1603 par Christoph Scheiner, un astronome allemand, mais l'outils fut réellement utilisé pour la typographie à partir du XIXe siècle avec le pantographe de Benton. Inventé par Lynd Boyd Benton, ce pantographie permettait de graver directement les poinçons à partir d'un dessin du glyphe. Une version du pantographe de Benton créé en 1899 et appelé le pantographe optomécanique permettait même par un jeu mécanique de pencher ou d'appliquer des transformations optiques à un dessin redimensionné par l'outil.

[ANN2] La lithographie

La lithographie est un technique d'impression utilisant le principe chimique du repoussement de l'eau et de la graisse. Inventée par Aloys Senefelder en 1796 en Allemagne, elle est véritablement utilisé à partir du XIXe siècle. Cette technique permet la création et la reproduction à

de multiples exemplaires d'un tracé réalisé à l'encre ou au crayon sur une pierre calcaire. L'évolution de cette technique entrainera la création de la chromolithographie puis de la technique offset.

[ANN3] Photocomposition

La photocomposition est une technique de composition de texte utilisant le principe photographique. À la différence de l'imprimerie traditionnelle (au plomb) les caractères ne sont plus en volume mais directement flashé sur une surface photosensible. La première photocomposeuse était la Lumitype, inventé par les ingénieurs français Louis Moyroud et René Higonnet en 1947. L'histoire de la photocomposition est très courte car elle opère la transition entre le typographie mécanique au plomb (linotype, monotype) et la typographie numérique. Néanmoins, plusieurs générations de photocomposeuses ont eu le temps de voir le jour. On peut en distinguer trois majeurs. La première génération permettait le flashage des types à travers un pochoir. La seconde avait recours à une décomposition des glyphes en bandes verticales afin de limiter les effets de la

réfraction. Et la troisième était entièrement numérique et un système de laser.

[ANN4] Encodage

À chaque type correspond un code permettant de le désigner. Originellement un code binaire fut assigné à chaque lettre de l'alphabet latin utilisé dans la langue anglaise. Par la suite ces codes se sont trouvés être trop limités pour couvrir l'ensemble des glyphes latins, puis non-latins. Des codes ISO furent ainsi mis en place localement. Mais afin d'être sûr qu'un même code affiche toujours la même lettre, un code universelle fut mis en place. Ce code est actuellement dirigé par un consortium, c'est l'Unicode. Un code unicode désigne donc un type, quelle que soit sa forme.

[ANN5] Fonte Bitmap

Une fonte bitmap est une fonte définie par un dessin formé de pixels. Les fontes bitmaps étaient réalisés pixels par pixels et destinées à un affichage à l'écran. De nombreuses fontes bitmaps furent notamment réalisés pour le lancement du Macintosh par Susan Kare. Assez rapidement elles furent accompagnées d'une fonte Postscript de Type 1 pour l'impression haute définition. Les fontes bitmaps étant dessinées pixel par pixels, elle ne peuvent être utilisées qu'au corps de texte pour lequel elles ont été

dessinés. Une fonte était donc dessinée pour chaque corps, faisant varier le dessin pour adapter la forme des glyphes manuellement à la grille de pixels. L'ensemble des fontes bitmaps et Postscript étaient alors contenues dans un même fichier appelé valise de fonte.

[ANN6] Postscript

En 1982, deux anciens membres du Xerox Parc, John Warnock et Charles Geschke (Paolo Alto Research Centre, cf. p. 165 de La typographie Moderne de Robin Kinross) créent l'entreprise Adobe pour commercialiser l'invention d'un nouveau langage de description de la page : le Postscript. Le Postscript révolutionna le monde de l'industrie graphique et informatique car il permit la standardisation de la description de la page pour les imprimantes. Jusqu'alors tous les terminaux d'impressions utilisaient des technologies propriétaires différentes rendant complexe voire impossible l'impression d'un document sans l'impression préalable de la typographie sur l'imprimante.

[ANN7] Postscript de type 1

Les formats Postscript de type 1 et 3, sont deux formats de fonte découlant du langage de description de page Postscript. C'est John Warnock, fondateur d'Adobe qui introduit ce système avec le format Postscript en 1983

(p.20 de Digital Typography & Artificial Intelligence). Le Postscript de Type 1 fut introduit sur Mac en 1985 et permit l'utilisation de fontes vectorielles, c'est à dire redimensionnable homothétiquement à toutes les échelles sans perte de qualité. Couplé au langage Postscript il permettait la création d'imprimé compatible avec toutes les imprimantes prenant en charge le langage. L'imprimante laser LaserWriter, lancé par Apple en 1985 fut l'une des première imprimante à le proposer et à en démocratiser l'usage. Une fonte PST1 pouvait contenir au maximum 256 caractères (2⁸). Les variantes telles que les petites capitales ou glyphes alternatifs étaient donc contenus dans d'autres fichier fontes.

[ANN8] Postscript de Type 3

Le Postscript de Type 3 est un format de fonte utilisant le langage Postscript lancé en même temps que le PST1. À la différence du format PST1, le PST3 est un format ouvert. Les fontes PST3 ne sont pas encapsulés dans des fichiers binaires illisibles, mais font parti intégrante du script Postscript décrivant la page. Ainsi le PST3 permet à l'utilisateur de créer ses propres fontes à l'intérieur même d'un fichier PostScript. À la différence d'une fonte PST1, l'ensemble des fonctions du langage Postscript

peuvent-être appliqué à une fonte PST3. Ainsi il devient possible d'appliquer des fonctions aléatoires ou des algorithmes pour définir le dessin des glyphes. Une fonte PST3 ne peut cependant être installée comme une fonte PST1 pour être utilisée en dehors d'un fichier Postscript.

Le PST3 fut peu utilisé, mais permis des expérimentations avec le format typographique digitale, telles que le projet Beowolf ou encore la version Postscript du Punk de Jacques André.

[ANN9] Truetype

Le format Truetype (TT) est un format de fonte qui a été développé par Apple en réponse au format PostScript de Type 1 développé par Adobe à la fin des années 80 et alors leader du marché. Tout comme le Postscript de Type 1, le Truetype est basé sur une description vectorielle des caractère et permet la mise à l'échelle des glyphes sans perte de qualité. Il pallie également certaines limitations techniques du format PostScript de Type 1, notamment dans le contexte d'affichage à l'écran. Microsoft a par la suite rejoint Apple dans le développement et l'utilisation du format. Le format a été implémenté sur Mac OS 7 en mai 1991 et sur Windows 3.1 en avril 1992. Lors de ses premières années, le format Truetype n'était pas compatible avec

le langage Postscript, ce qui amena les designers graphiques à lui préférer le format Postscript Type 1, déjà largement implanté. Le TT resta ainsi un certain temps circonscrit au rôle de fonte d'affichage. Ce problème de compatibilité a par la suite été résolu. À la différence du `PST1`, le format TTF n'utilise pas les courbes de Bézier (cubique), mais des courbes quadratiques. Cette particularité, couplé à la fonction de hinting permet une meilleure interprétation des courbes à travers la grille de pixels (voir Courbes de Bézier cubique, Courbes de Bézier quadratique et Hinting). Ainsi, à la différence du format `PST1`, ce format ne nécessitait pas l'usage de fonte bitmap en complément de la fonte vectorielle pour l'affichage sur écran. De plus l'utilisation de courbes de Béziers quadratique permet de réduire le nombre de points d'ancrage. Le format TTF est donc plus léger que le format `PST1`, ce qui fut un avantage dans son utilisation dans un contexte web. Cependant aujourd'hui les formats spécifiques car optimisés pour le web (`WOFF` et `WOFF2`) lui sont préférés dans ce contexte. Une version du format TTF a été réalisée par Microsoft en 1995 en réponse au format Truetype GX d'Apple : le TrueType Open. Ce format donne au format TTF la capacité de gérer l'utilisation automatisé

de caractères alternatifs et de substitutions de glyphes. Ce format plus simple mais moins complet que le Truetype GX, évoluera par la suite en partenariat avec Adobe et Apple, en 1997, pour donner le format OpenType TTF.

[ANN10] Hinting

Le Hinting est une manière d'améliorer le tramage des tracés vectoriels dans la grille de pixel pour les différents corps et les différentes résolutions d'affichage. Il se concrétise par une série d'instructions s'ajoutants à la description vectorielle des glyphes. Ces instructions permettent de modifier les courbes des glyphes pour permettre leur correction optique dans la grille de pixels. Le hinting est apparu avec le format Truetype et a permis l'utilisation de fontes vectorielles à l'écran sans avoir recours à des fontes bitmaps créées manuellement. Aujourd'hui il est possible d'utiliser le hinting avec le format OpenType, qu'il s'agisse de courbes cubiques ou quadratiques. Cependant les instructions à donner ne seront pas du même type dans les deux cas.

[ANN11] Truetype GX

Le format Truetype a été étendu par Apple en 1995 pour devenir Truetype GX. Cette extension du format consistait en une mise à niveau de ce dernier sur

l'architecture Quickdraw GX implémentée dans le système d'exploitation Mac OS (voir QuickDraw GX). Le format s'est alors vu supporter des fonctionnalités de variations stylistiques, de substitutions contextuelles de gestion des lignes de texte et d'interpolation live des glyphes.

[ANN12] **Format Multiple Master**
Le format Multiple Master est un format permettant l'interpolation de plusieurs dessins maitres contenus dans un même fichier fonte. Il fut publié par Adobe en 1991. Dans sa forme logicielle, le format MMF est une extension du format Postscript de Type 1. Il n'y ajoute d'ailleurs aucune autre fonctionnalité mis à part le support de plusieurs masters (dessins maitres) dans un même fichier. Adobe a arrêté de créer et de supporter le format Multiple Master en 1999, et même si les premières versions d'essai du format OpenType intégrèrent le support des multiples masters, ceux-ci furent abandonnés bien avant le lancement du format OpenType..

[ANN13] **Courbes de Bézier**
Créée dans les années 1950, la courbe de Bézier est un objet mathématique qui fut inventée par Pierre Bézier dans le cadre de son travail en tant qu'ingénieur pour les usines Renault. Jusqu'alors, la description des formes

des carrosseries et autres pièces en volumes était réalisée à travers des montages de courbes matérielles analogiques [Image et réf.], rendant la forme peu précise, complexe à mettre en place et à partager. Afin de répondre à ce problème, Pierre Bézier inventa une courbe mathématique basée sur des points d'ancrage manipulables aux coordonnées définies sur un plan orthonormé. Bien entendu, Pierre Bézier n'est pas le seul à avoir travaillé sur ces fameuses courbes. Un des autres grands contributeurs à cette invention est Henry de Casteljau qui traduisit le concept de la courbe de Bézier à travers un algorithme : l'algorithme de Casteljau.

[ANN14] **Algorithme de Casteljau**
Paul de Casteljau est l'inventeur de cet algorithme utilisé pour dessiner des courbes et des surfaces de Bézier et aujourd'hui au cœur de tous les logiciels de reproduction, d'affichage et de création graphique. Avec le recours à la courbe de Bézier, et donc à l'algorithme de Casteljau, le dessin devient entièrement numérique. Cet algorithme repose sur une formule simple répétée en boucles : l'interpolation. La voici : Résultat = Valeur 1 + Facteur x (Valeur 2 - Valeur 1)

[ANN15] Courbe quadratique
Il existe plusieurs types de courbes de Bézier. La plus simple mathématiquement est appelée la courbe de bézier quadratique. Pour la définir, il faut tracer deux segments reliés entre eux par un point. On réalise alors une interpolation pour chaque segment en prenant pour valeur chaque extrémité du point et en faisant ensuite varier le facteur de zéro à un. Une fois cela fait il ne reste plus qu'à réaliser une seconde interpolation basée sur les valeurs trouvées pour chacun des segments et chacun des facteurs. En calculant l'ensemble des valeurs, on obtient alors un ensemble de points suivant une courbe. Plus le nombre de calculs est important, plus le dessin de la courbe sera fin. Nous obtenons ainsi une courbe de bézier. Par analogie, on peut donc dire que le point, notre outil — que l'on pourrait définir rond, carré ou toute autre forme — se trouve déplacé dans un mouvement de translation par une fonction mathématique. échelle.

[ANN16] Courbe cubique
Il existe également un autre type de courbe de bézie, la courbe de bézier cubique. C'est celle ci que nous utilisons pour dessiner sur nos interfaces graphiques. Bien que plus complexe dans ses mathématiques (elle résulte

de trois interpolation imbriquées, elle est beaucoup plus simple d'usage, car plus intuitive. Pour autant la courbe quadratique est toujours utilisée de nos jours. Nous reviendrons sur cet aspect un peu plus tard.

[ANN17] OpenType
Le format OpenType a été créé en 1996 par Adobe, Microsoft. Ce format tente de réunir le meilleur des deux mondes Postscript et TrueType en permettant l'encapsulation de courbes cubiques (Postscript) et quadratiques (TrueType). Il inclut les variations et substitutions introduites par le format TrueType Open et permet de contenir plus de 65 000 caractères, mettant fin à la limitation de 256 glyphes imposée par le format Postscript de Type 1. Ce format est aujourd'hui le standard en typographie. Il est très largement utilisé et distribué par les fonderies de caractères. Ces spécifications de font de lui un format idéal pour répondre à des contextes d'affichage à l'écran (avec l'utilisation de courbes quadratiques) et d'imprimé (courbes cubiques). Il se voit néanmoins petit à petit concurrencé sur le web par le jeune format web WOFF (Web Open Font Format) et WOFF2 qui lui sont préférés en raison de leur meilleure compression.

[ANN18] Python

Le Python est un langage de programmation. Sa particularité est d'imposer une mise en forme du code extrêmement stricte à son utilisateur. Ainsi, en python chaque chose ne devrait être réalisable que d'une et une seule manière. En effet, la mise en forme du code fait parti intégrante du code lui même. Si le langage Python s'est trouvé aussi facilement utilisé par la communauté des designers typographiques, c'est également en raison de la simplicité de sa syntaxe. En effet, ce langage de programmation a été conçu pour être un langage visant à être visuellement épuré et facile d'accès. Il s'inscrit dans la même veine que le Smalltalk, langage développé au Xerox Parc avec l'intention de rendre le code plus accessible en utilisant le langage courant. La facilité d'accès et la rapidité de mise en forme du python en a rapidement fait un langage de script. C'est à dire un langage permettant d'écrire de petits codes pour automatiser des tâches simples, mais fastidieuses. Pour cette raison le python est un langage parfaitement adapté à la création typographique. Cependant si le python est aussi utilisé dans le monde de la typographie c'est également parce qu'il a été créé en 1990 par Guido Van Rossum, le frère de Just Van Rossum, l'un des

deux designers du collectif Letterror, très actifs sur la scène typographique numérique et professeurs au Type and Media de l'Académie Royale de La Haye.

*"Explicit is better than implicit
There should be one- — and preferably only one -- obvious way to do it."*

[The Zen of Python]

[ANN19] Robofab

Cette librairie open source est née en 1996 avec Robofog. Une extension de Fontographer 3.5 (le logiciel typographique dominant alors le marché) implantant la première version de Robofab. En permettant la manipulation de l'objet fonte à travers tous ces niveaux via la programmation, Robofab donna accès aux designers typographiques à la puissance de calcul de l'ordinateur et à l'automatisation des tâches. La maîtrise de la programmation dans le processus de création d'une fonte digitale est une composante importante. Au delà de la facilité et de la productivité offerte par le recours à un langage de programmation à même d'automatiser des tâches répétitives (qui sont nombreuses dans le cas du design typographique, la programmation permet la manipulation et la création de règles algorithmiques permettant la modification ou la création de points de la

courbe de bézier.

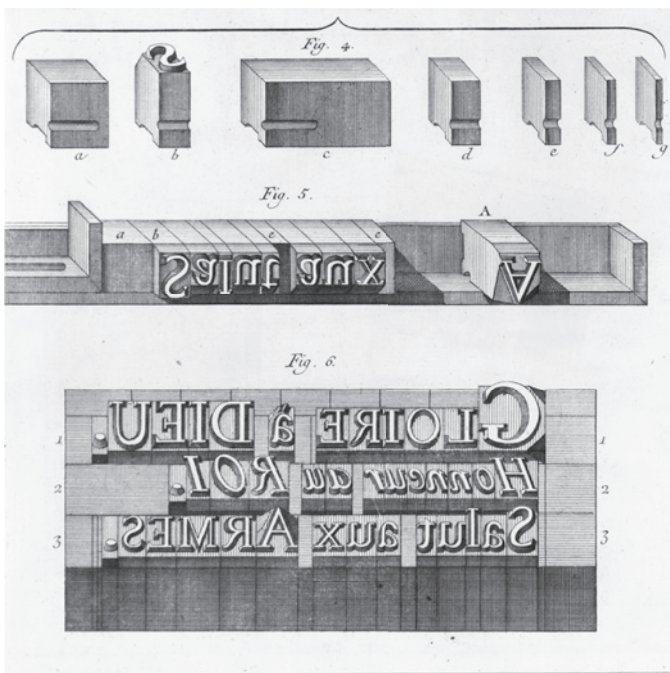
"Because my circle is maybe different of your circle."
[Frederik Berlaen]

[ANN20] Environnement graphique d'aide à la programmation C'est dans l'idée de permettre ce type de création qu'a été créé Robofon, mais également Drawbot, un espace de programmation graphique en python. Composé d'un espace de live coding, d'un espace de visualisation graphique et d'une console d'output, drawbot est basé sur le moteur de rendu de PDF de Mac Os (Quartz). Cela lui permet ainsi d'avoir accès aux fonctionnalités opentype et à la gestion typographique fine de Mac Os X. Cet éditeur libre de droit a été créé par Frederick Berlaen (créateur de Robofont) et Just Van Rossum (un des créateur de Robofab et membre de Letterror). Extensibl, il est aujourd'hui pourvu de la librairie Robofa, d'un équivalent du Postscript et de librairie d'export vers le SVG, le PDF, PNG et JPG. Il s'agit là d'une belle amorce pour un éditeur de mise en page libre et modulable.

[ANN21] L'Ikarus Mouse
Le transfert du dessin de caractères sur du papier calque à son dessin directement sur écran ne s'est pas fait de façon direct. Un outils intermédiaire fut utilisé. Véritable chainon manquant

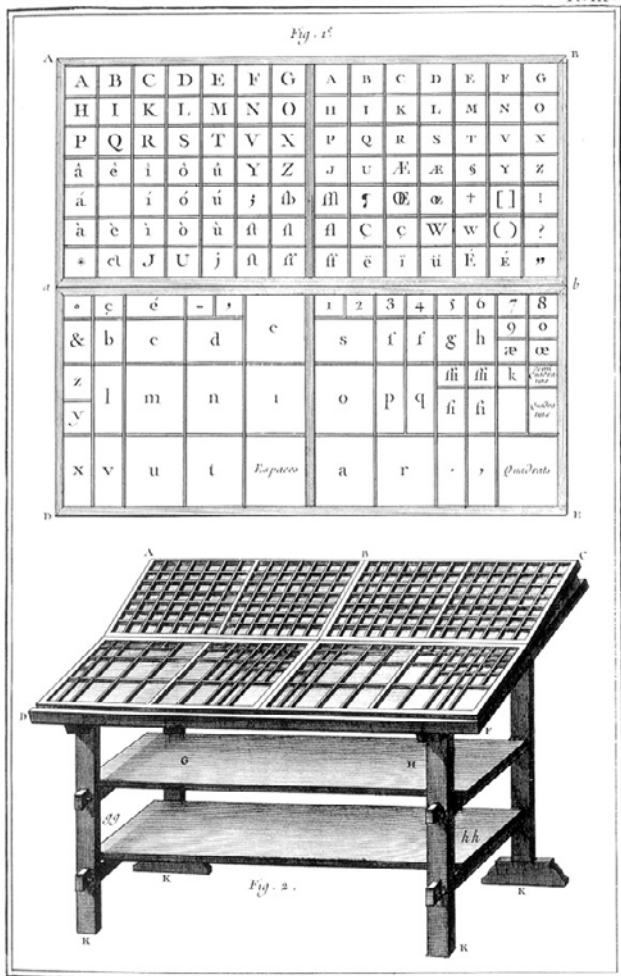
de la chaîne de production typographique, l'Ikarus mouse a longtemps permis la vectorisation des fontes du monde de la photocomposition au monde du digital. Véritable scanner vectoriel manuel, cet outil est, en quelque sorte l'ancêtre des script d'autotrace (des logiciels permettant la reconnaissance du tracé central d'un dessin) qui virent le jour de nombreuses années plus tard. Cet outil permet de mieux comprendre la complexité de réalisation d'une fontes digitale destiné à l'impression aux balbutiements de l'informatique. En effet la technologie des écran ne permettant pas une résolution suffisante il était alors complexe d'obtenir des tracés d'une qualité suffisante en se fiant uniquement à l'écran. De plus, la majorité des fontes étant alors présente sous le format de la photocomposition, il était indispensable de permettre leur conversion au format digital, ce que cet outils permit.

[A, 1—6]
Évolution de la composition
Composition au plomb



[A1]

Encyclopédie de Diderot et d'Alembert,
 Imprimerie, Reliure / Diderot et d'Alembert, 1751-1780



Goussier Del.

Benard Fecit.

Imprimerie, Casse

[A2]

Encyclopédie de Diderot et d'Alembert,
Imprimerie, Reliure / Diderot et d'Alembert, 1751-1780



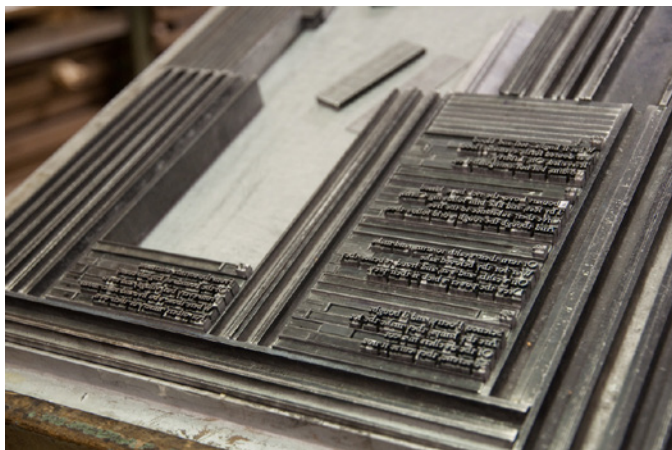
1949

Atelier de composition (Imp. Alençon-
1949)

9^e (file de droite)

[A3]

Atelier de composition typographique à Alençon, Anonyme, 1949



[A4]

Types de plomb composés, Photographie libre de droits



[A5]

Casse typographique, Photographie libre de droits



[A6]

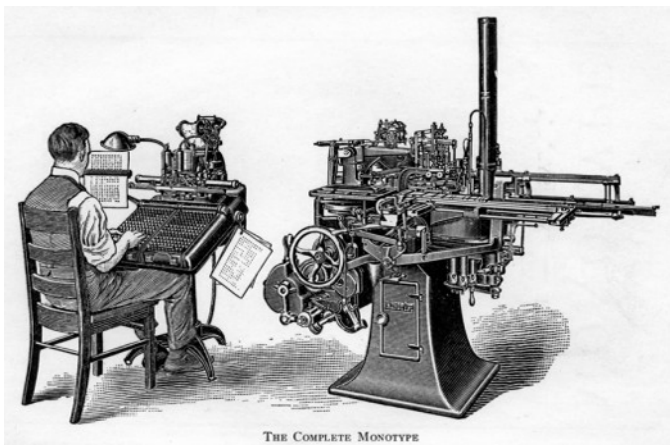
Typographe, vue sur la gallée de la presse typographique,
Ouest France, Nicolas Moutard

[A, 7 — 19]
Composition mécanique



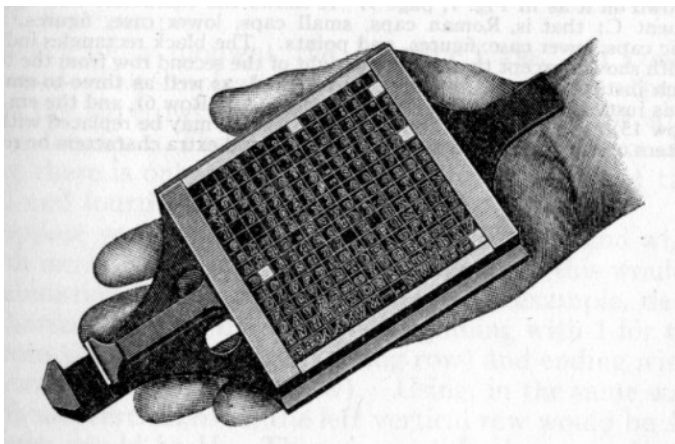
[A8]

Opératrice, Lanston Monotype Corporation, 1924



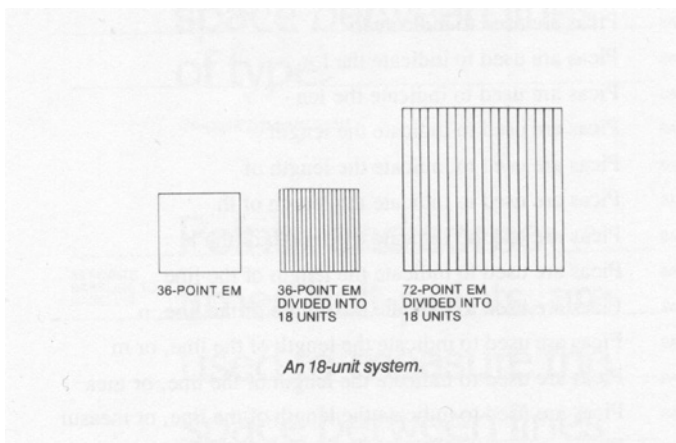
[A9]

Opérateur et Monotype, Lanston Monotype Corporation, 1916



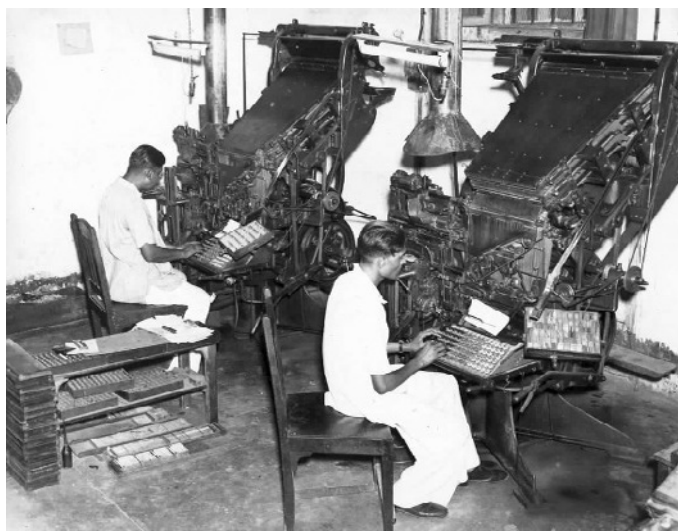
[A10]

Matrice Monotype, Lanston Monotype Corporation, 1924



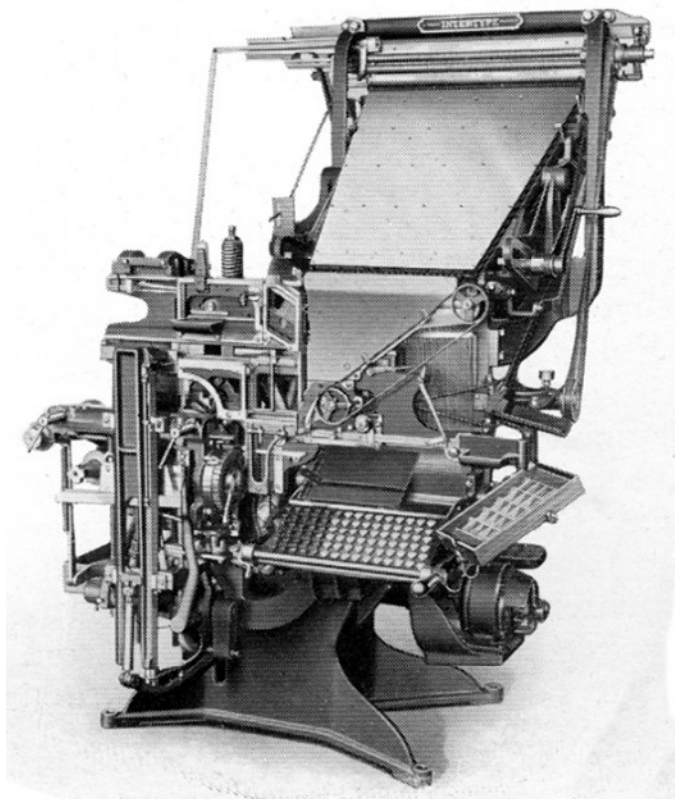
[A11]

Phototypesetting, A design manual, James Craig, 1978



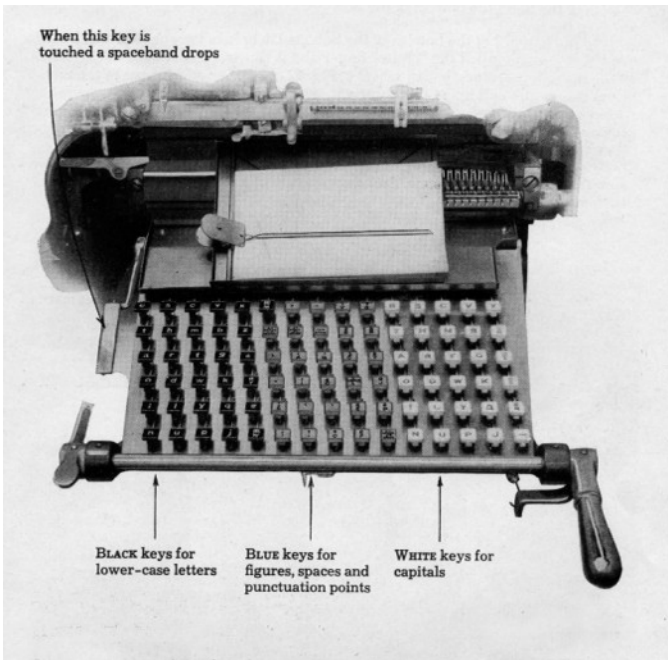
[A12]

Linotype, Linotype Corporation



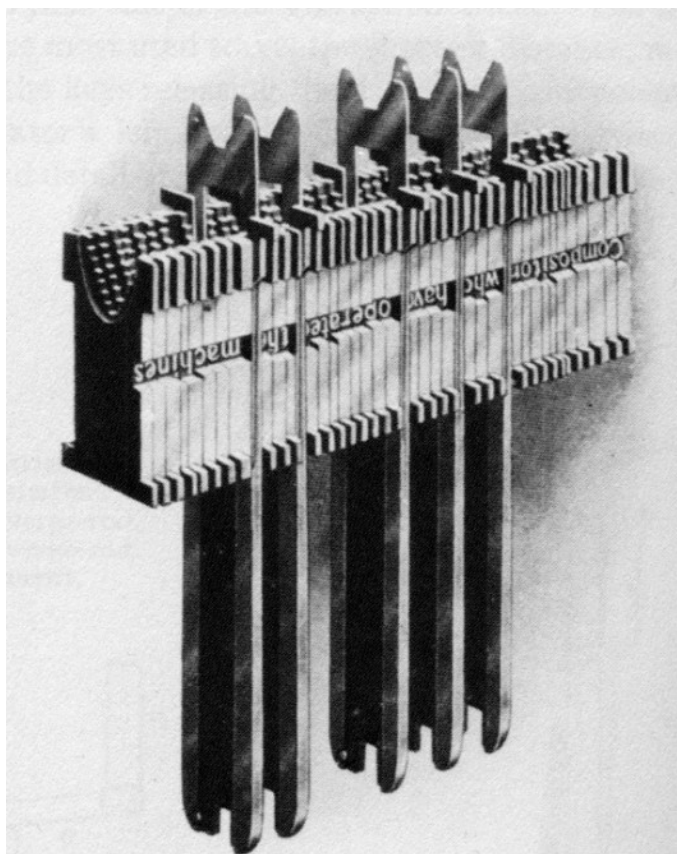
[A13]

Intertype Model C, Commercial engraving and printing,
Linotype Corporation, 1924



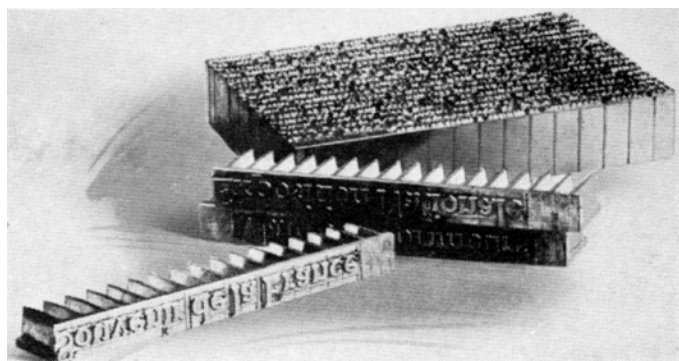
[A14]

Clavier Linotype, Dr. David MacMillan, Linotype Corporation



[A15]

Types de Linotype, Commercial engraving and printing,
Linotype Corporatio



[A16]

Ligne de plomb Linotype, Commercial engraving and printing,
Linotype Corporation



FIG. 11-1. Simplicity of handling has established the Linotype slug as the modern unit of composition.

[A17]

Composition de lignes Linotype formant un bloc,
Linotype Corporation



Set-width, or unit value, for three characters.

| CHARACTERS | SET-WIDTH |
|--|-----------|
| i, j, l | 4 |
| f, t, l, ., , | 5 |
| r | 6 |
| c, k, s, v, x, y, z, J | 9 |
| a, b, d, e, g, h, n, o, p, q, u, L 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 | 10 |
| F, T, Z | 11 |
| A, B, E, K, P, V, X, Y | 12 |
| w, C, D, H, N, U, R | 13 |
| G, O, Q | 14 |
| m, M | 15 |
| W | 17 |

*Set-width, expressed in units, for 12-point
Helvetica Roman.*

[A18]

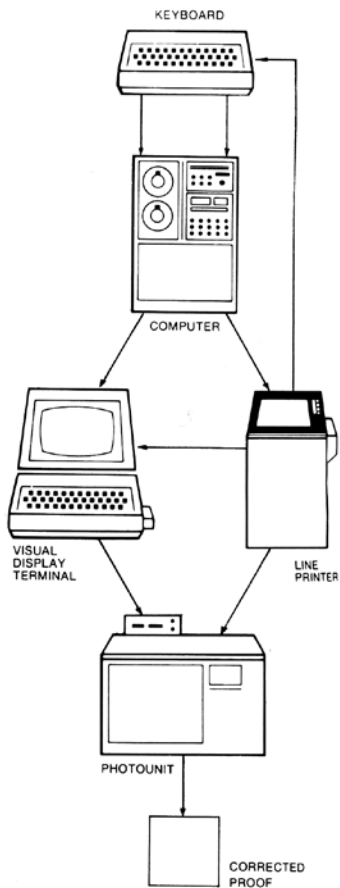
Phototype setting, A design manual, James Craig, 1978



[A19]

Machine à écrire Valentine par Olivetti, Ettore Sottsass, 1969

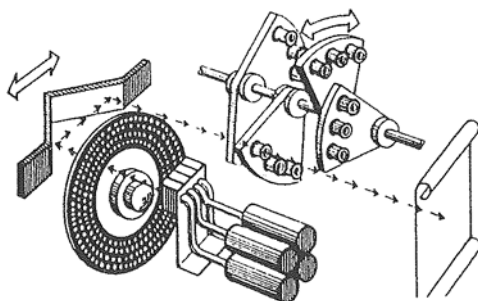
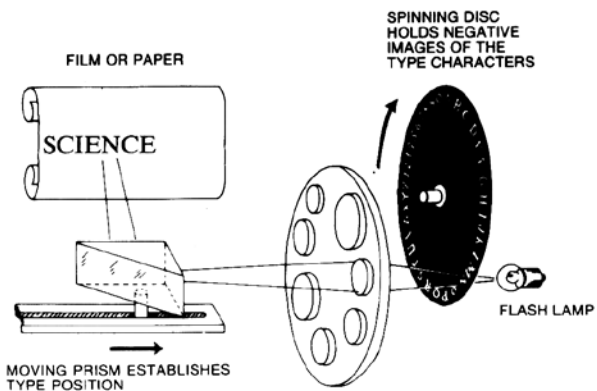
[A, 20—24]
Photocomposition



Editing and correcting sequences using either a VDT or a line printer.

[A20]

Phototypesetting, A design manual, James Craig, 1978

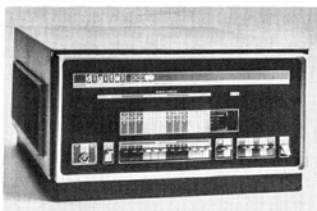


[A21]

Images extraites de Alice Savoie [BIBLIOGRAPHIE 19]
et Peter Karon [BIBLIOGRAPHIE 3]



Keyboard, for input.



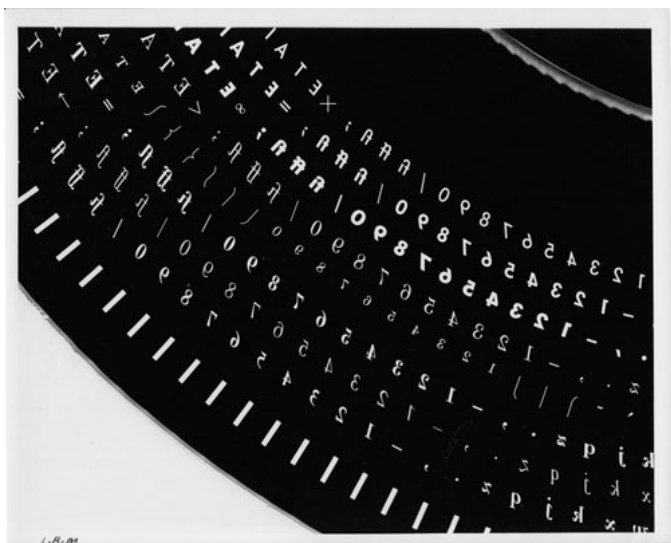
Computer, for making end-of-line decisions.



Photounit, or typesetter, for output.

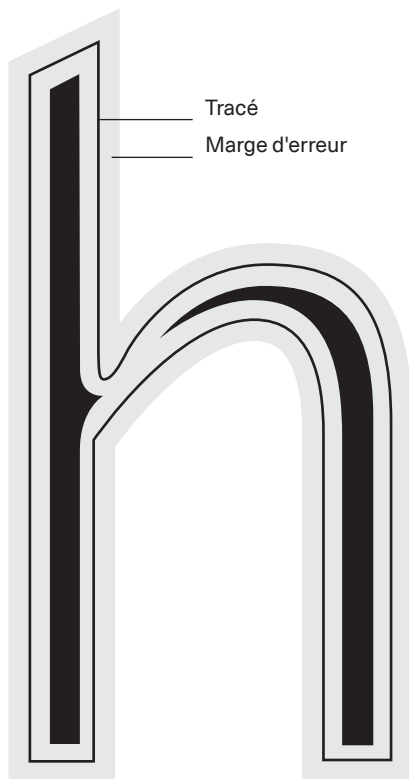
[A22]

Phototypesetting, A design manual, James Craig, 1978



[A23]

Image extraite de Alice Savoie [BIBLIOGRAPHIE 3]



[A24]

Marge d'erreur lors d'un flashage en photocomposition

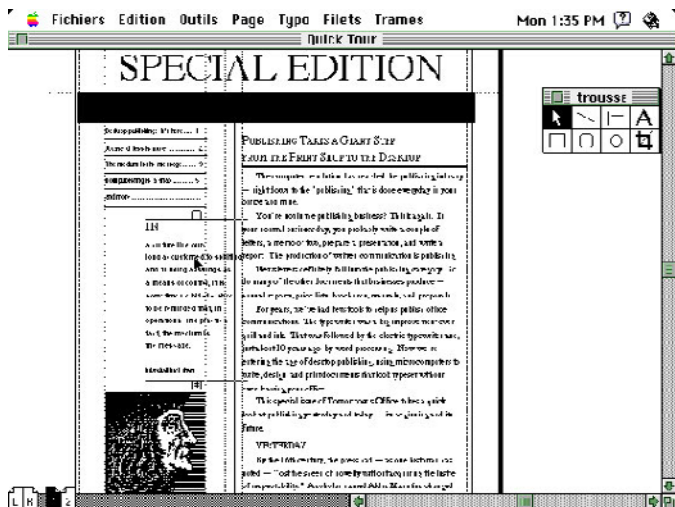
[A, 25—29]

Composition numérique



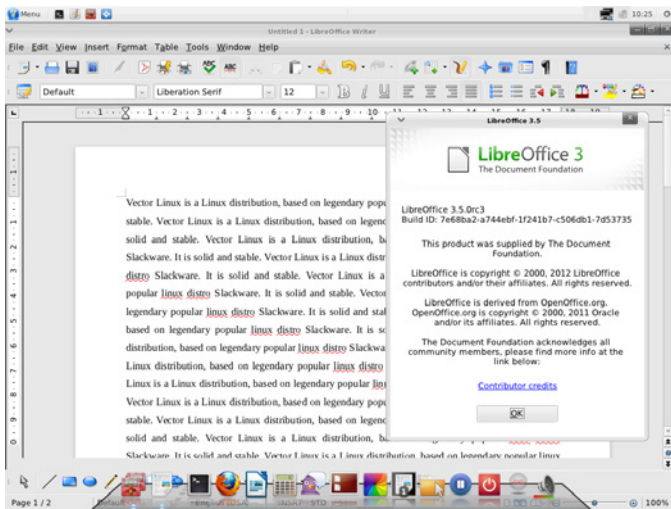
[A25]

Hypertext editing System à la Brown University, Greg Lloyd, 1969



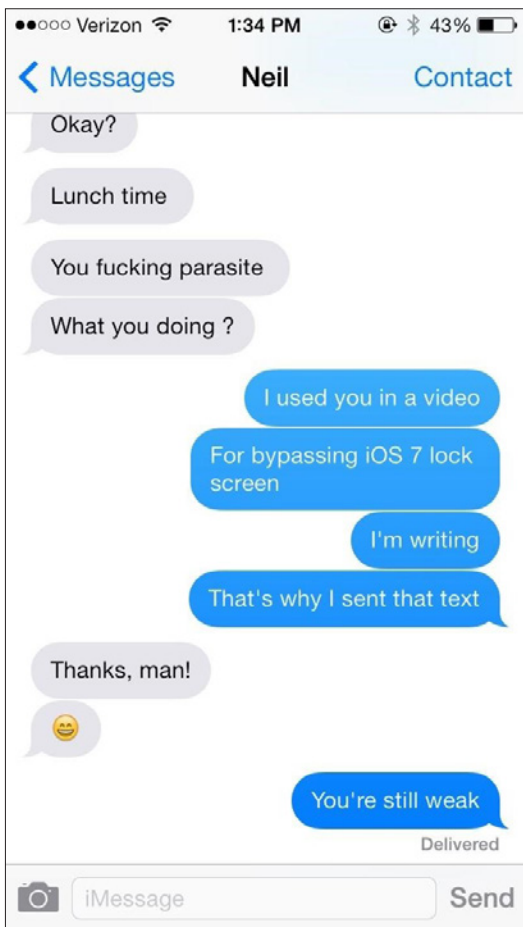
[A26]

Capture d'écran de Page Maker version 1 sur Mac OS, Aldus, 1985



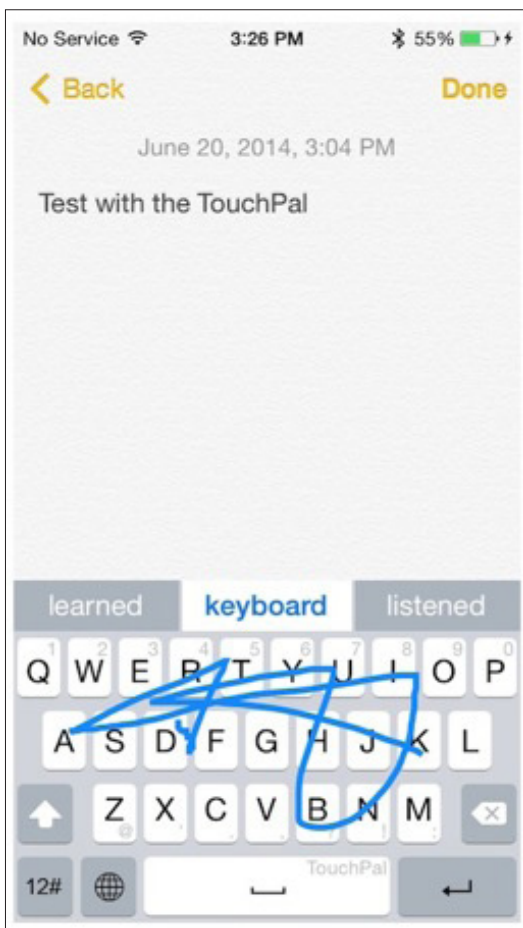
[A27]

Capture d'écran de Libre Office 3 sur Linux, 2015



[A28]

Capture d'écran iMessage IOS, 2015



[A29]

Capture d'écran clavier IOS, 2015

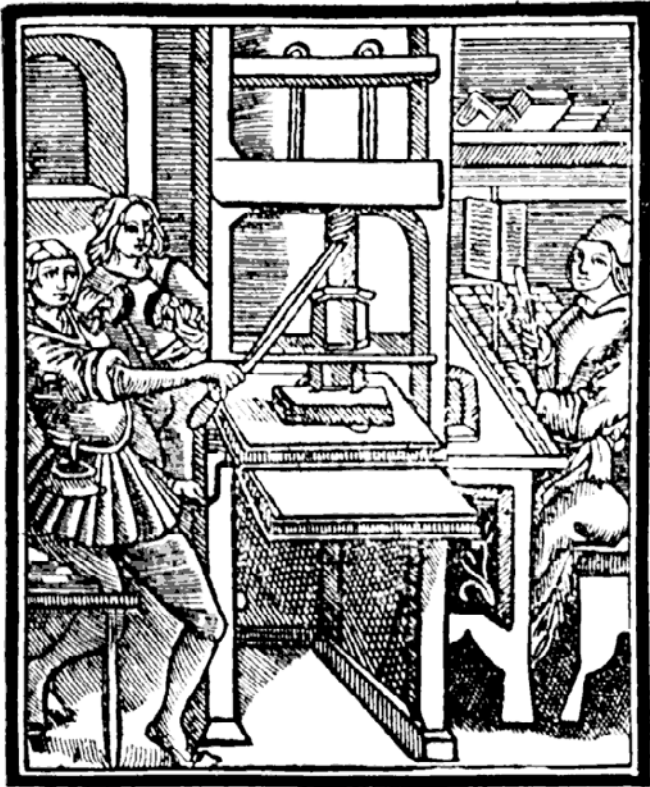
[B, 1 — 6]

Les dispositifs d'affichages
Impression et papier

iudam: anno quarto achab regis israhel. Trigintaquinque annos erat cum regnare cepisset: et vigintiquinque annis regnavit in iherusalem. Mometi matris eius azuba: filia salai. Et ambulavit in omni via asa patris sui: et non dedinavit eam. Fecitque quod rectum est in conspectu domini: veruntamen excelsa non abstulit. Adhuc enim populus sacrificabat et adolebat incensum in excelsis. Pacemque habuit iosaphat cum rege israhel. Reliqua autem verborum iosaphat. et opera eius que gessit et prelia nomine hec scripta sunt in libro verborum dierum regum iuda. Sed et reliquias effeminatorum qui remanserant in diebus asa patris eius abstulit de terra. Nec erat tunc rex constitutus in edom. Rex vero iosaphat fecerat classes in mari. que navigarent in ophir propter aurum: et ire non poterant: quia confracte sunt in aliyongaber. Tunc ait othzias filius achab ad iosaphat. Vadant serui mei cum servis tuis in navibus. Iste voluit iosaphat

[B1]

Extrait de la Bible 42 lignes de Gutenberg, Gallica, 1450



[B2]

Gravure de la manipulation d'une presse à bois dans un atelier typographique au 17e siècle, image libre de droits, Wikimedia

AVLI GELII NOCTIVM ATTIC
CAPITVLA PRIMI LIBRI.

Vali proportione: quibusq; collect
esse Pythagorā philosophū dixerit
proceritate: qua fuit Hercules: cum uita

Caput primum. Plutarchus.

Ab Herode Attico consulari uiro tempo
iactatum & gloriosum adolescentē speci
uerba Epicteti stoici: quibus festiuiter a
loquacium nebulonum: qui se stoicos n

Caput secundum. Herodes Atticus.

Quod Chilo lacedemonius consilium a
Quodque est circumspecte & anxie conf
amicorum delinquendum aliquando sit

Theophrastus: & M. Cicero super ea re

Caput. iii. Chilo præstabilis.

[B3]

Typographie Humanistique, source inconnue



[B4]

Plaque offset, Image libre de droits

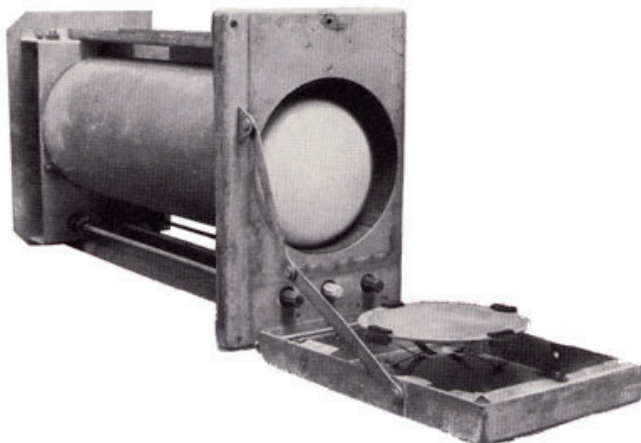
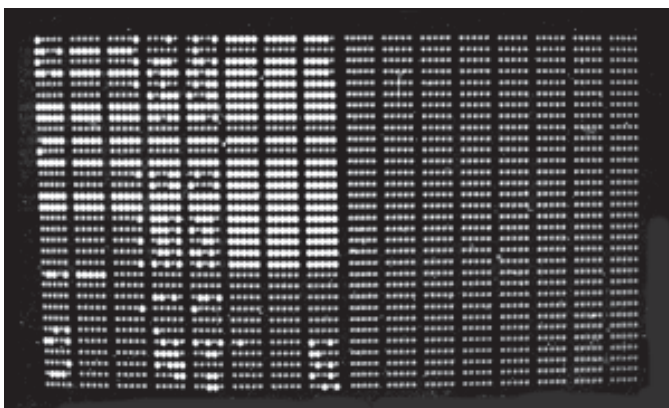


[B5]

Apple LaserWriter, Museum Victoria, 1987

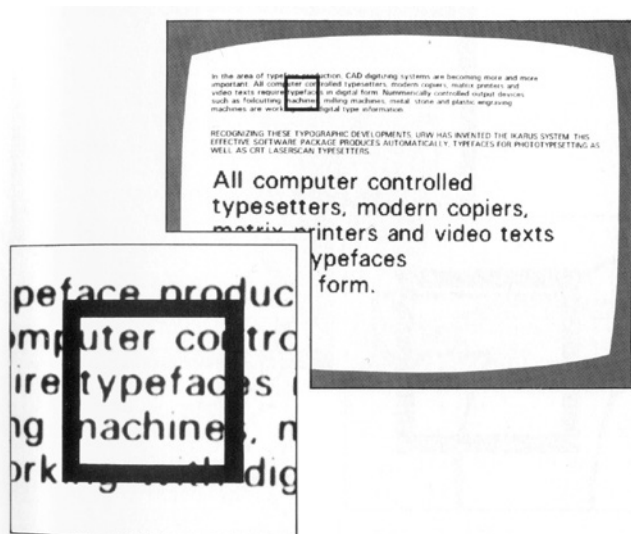
[B, 7—15]

Écrans



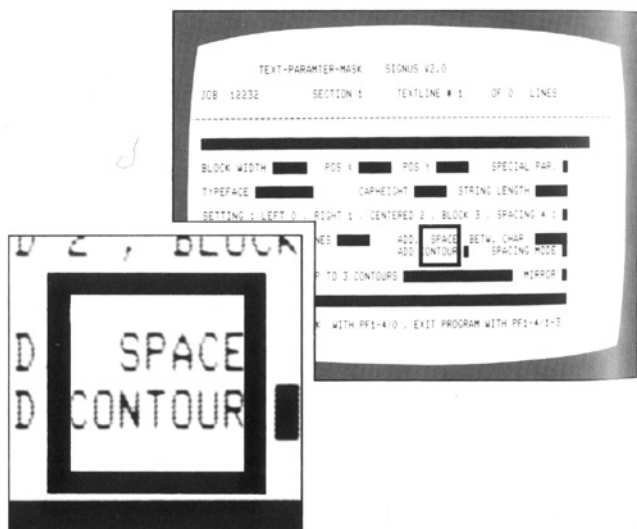
[B7]

Tube de Williams, Williams Kilburn,
Musée virtuel de l'informatique, 1946



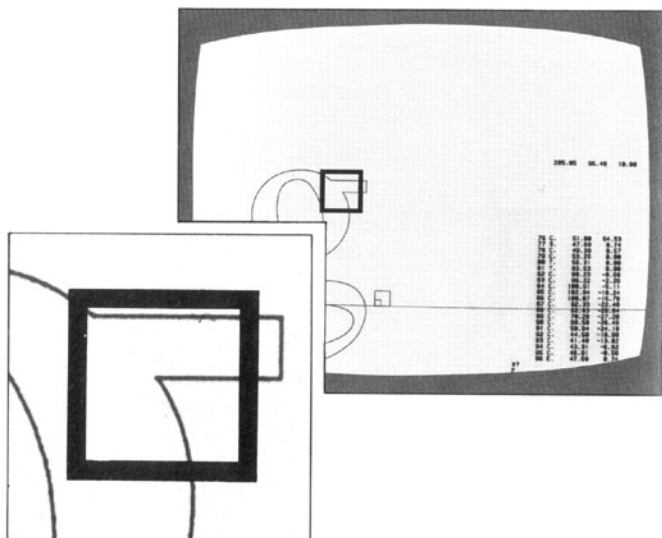
[B8]

Écran CRT 1^{ère} génération, Peter Karrow [BIBLIOGRAPHIE 3]



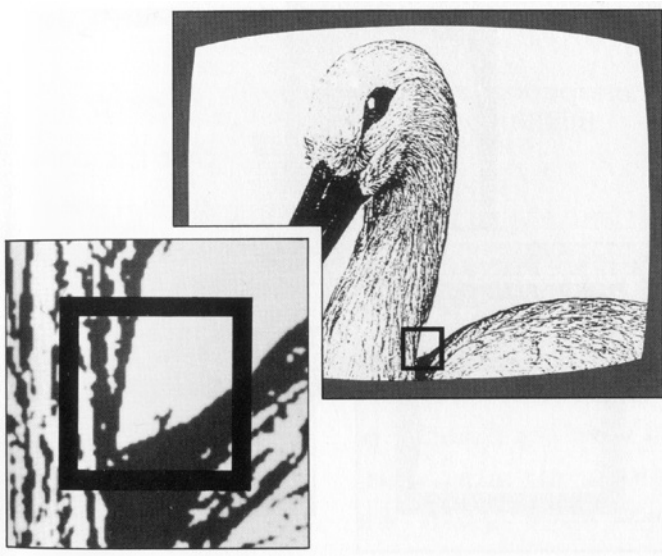
[B9]

Écran CRT 1^{ère} génération, Peter Karrow [BIBLIOGRAPHIE 3]



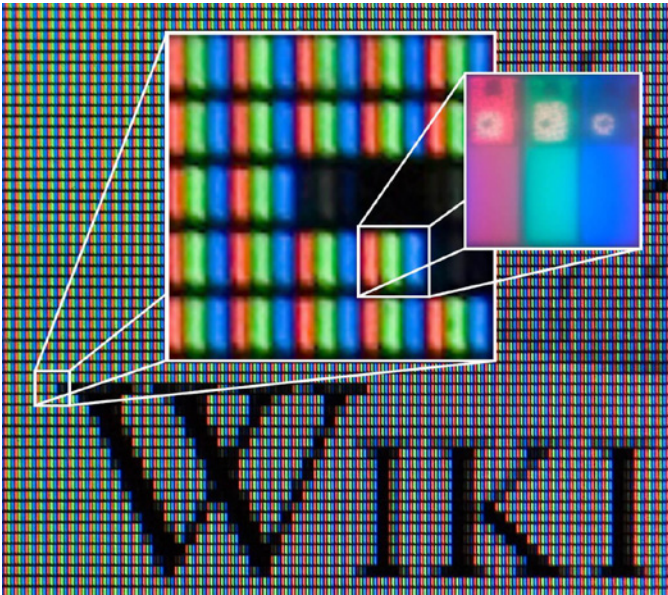
[B10]

Écran CRT 2^e génération, Peter Karrow [BIBLIOGRAPHIE 3]



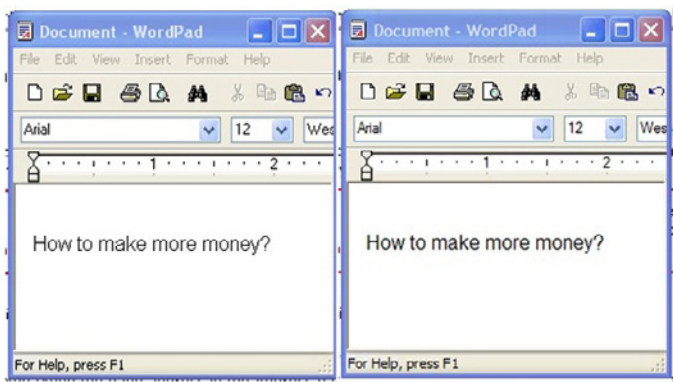
[B11]

Écran CRT 2^e génération, Peter Karrow [BIBLIOGRAPHIE 3]



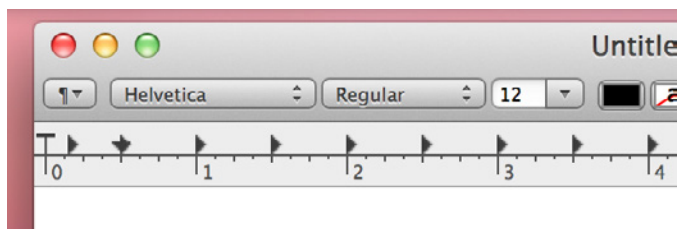
[B12]

Zoom sur une dalle LCD, Image libre de droits



[B13]

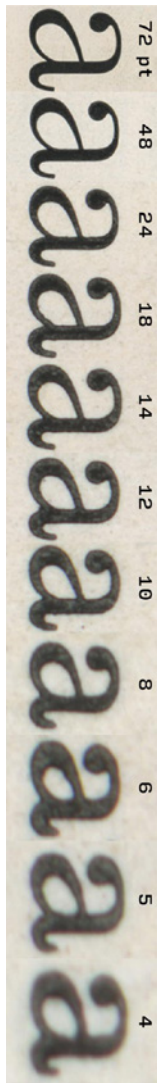
Capture d'écran, technologie de rendu sous pixels
Cleartype sur Windows XP



[B14]

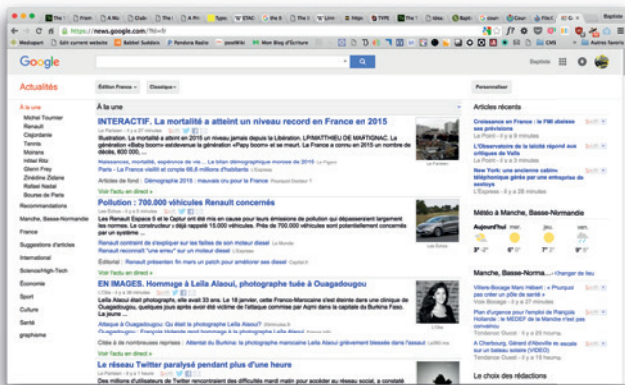
Capture d'écran, Évolution de la résolution d'affichage
sous Windows 3.1 (1992) et Mac OS X 10 Retina (2014)

[B, 15 — 19]
Échelles de lecture



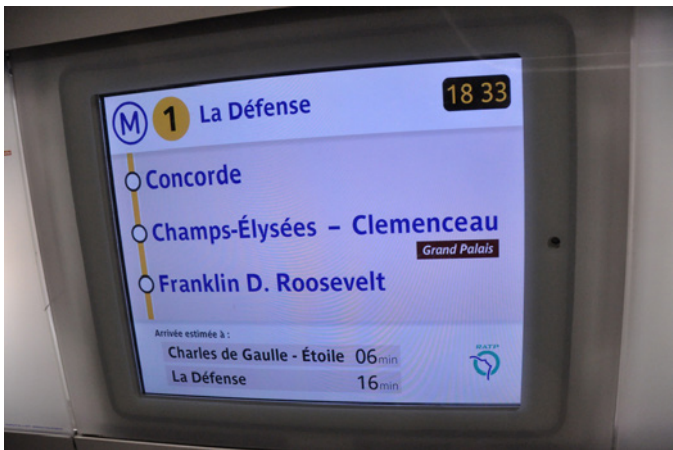
[B15]

Corps du Century Expanded plomb de 72 pt à 4 pt
remis à une même échelle, Benton Modern



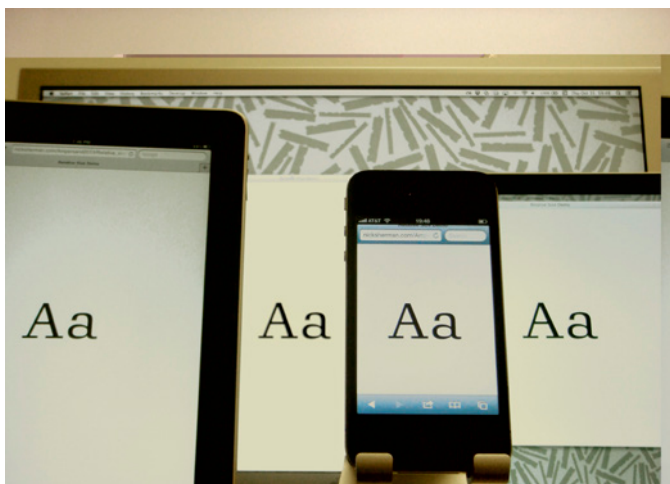
[B16]

Design adaptable, Capture d'écran d'une page Google News sur mobile et ordinateur portable



[B17]

Vue sur un écran de métro ligne 1 à Paris, Image libre de droits



[B18]

Vue de différents dispositifs d'affichage, Nick Sherman
[Webographie 28], 2014



[B19]

Vue d'un panneau d'affichage, Image libre de droits

[c, 1 — 10]
Création de caractères
Rationalisation et géométrisation de la lettre

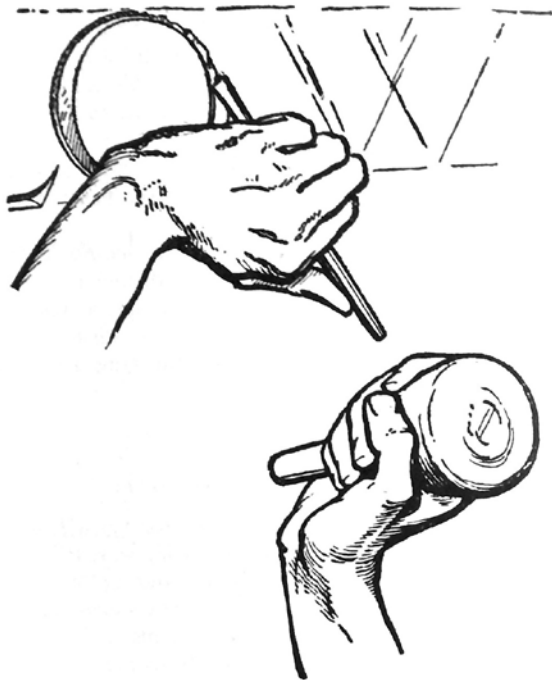
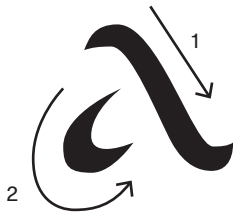
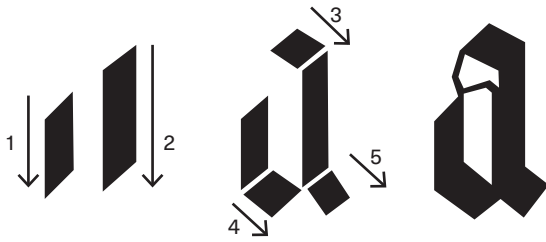


FIG. 214.

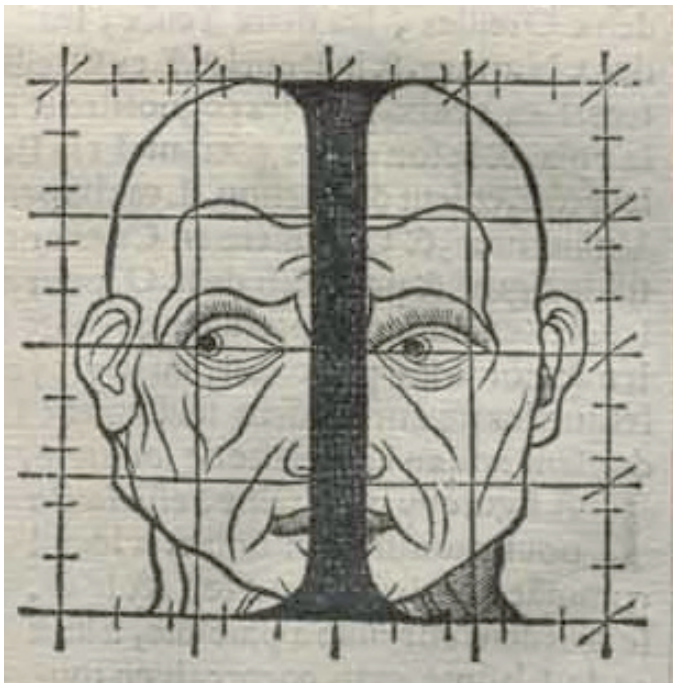
[C-1]

Gravure, Eric Gill, Edward Johnston,
Writing & Illuminating & Lettering, Lettered Matter



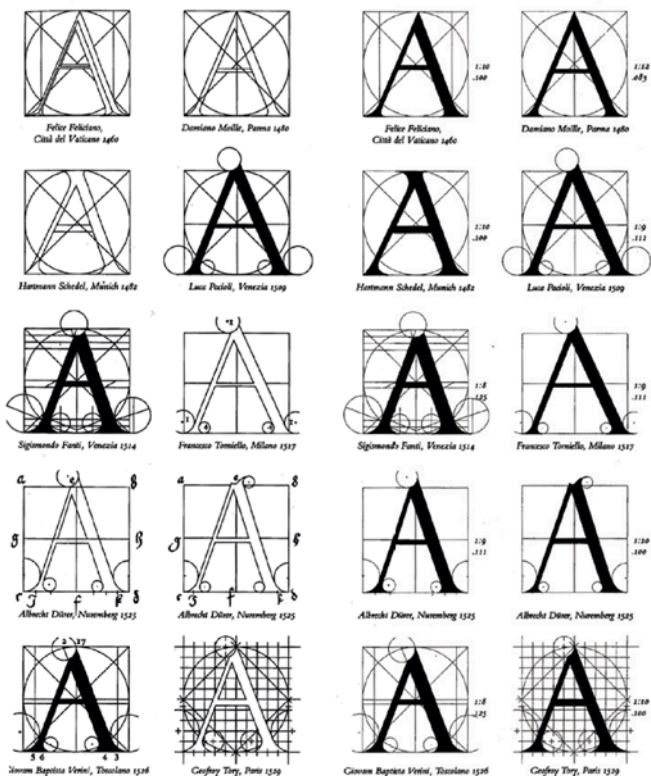
[c2]

Ductus calligraphique: Gothique Textura et Caroline



[c3]

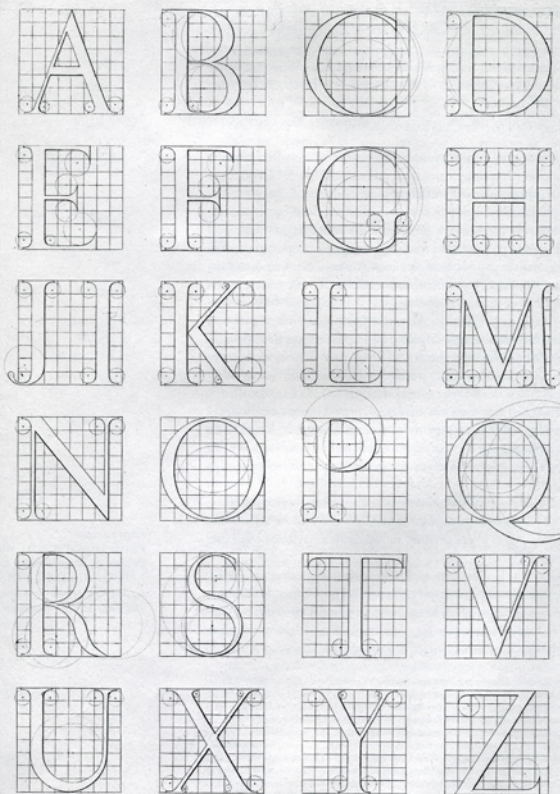
Geoffroy Tory, 1529



[C4]

Collection de douze A construits datants de 1460

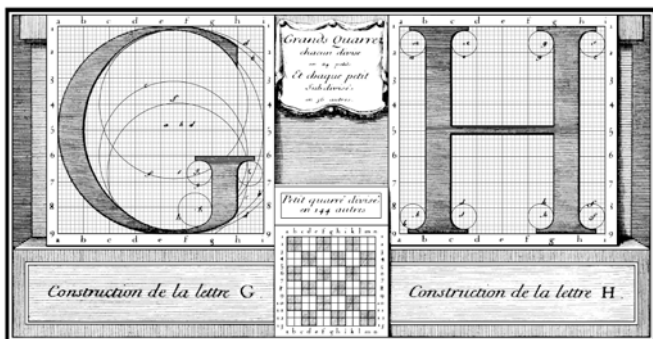
*Construction des Lettres
Capitales Droites.*



L'Édition Simenon, 1955

[C5]

Le Romain du Roi, La typographie au service de l'État,
1702-2002, ville de Lyon



[C6]

Le Romain du Roi, La typographie au service de l'État,
1702-2002, ville de Lyon

GRAND SUCCÈS DE L'EXPOSITION AMÉRICAINE

Plaque Découpée Universelle DE J. A. DAVID

BREVETÉ EN FRANCE ET À L'ÉTRANGER
POUR TRACER GÉOMÉTRIQUEMENT
LES LETTRES, LES CHIFFRES & LES ORNEMENTS

Le savez-vous déjà, il paraît certain que les Chinois se servent de ce mille créateur pour leurs ouvrages.

L'inventeur du dessin ci-contre, M. J. A. David, trouve abstrait en quelque sorte à son caractère géométrique, il veut tracer d'une façon simple et géométrique, les lettres, les chiffres et les ornements, les ensembles modernes, toutes celles qui dérivent du grec et du latin, telles que : le temple, l'arcade, l'ionique, l'égyptien, etc. Il trace en outre tous les chiffres, tous les ornements, tous les signes de sonnettes, sans qu'il y ait véritablement d'ensemble géométrique avec les figures doubles ou simples. L'Alphabet grec se trace aussi avec ce type.

Avec l'aide de cette plaque et de la vier géométrique qui l'accompagne, toute personne, même un enfant intelligent peut de suite sans peine apprendre en quelques heures à tracer tout allégrement les lettres et chiffres.

Elle est indispensable pour tracer les lettres dans tous les alphabets. Tout homme ayant à faire des ouvrages dérivés ou pour tracer, etc., dans cette simple plaque, il a un cadre complet du dessin de la lettre et des chiffres. Il apprendra, en outre, à tracer une quantité infinie de dessins pour bandes, bordures, gazons, etc.

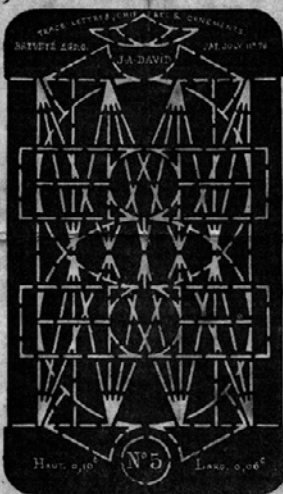
Dans la feuille il sera un jeu d'apprendre et de tracer les lettres, chiffres et les ornements, car, en outre des lettres, des chiffres et des ornements, ils y trouvent tous les éléments du dessin linéaire.

La plaque découpée universelle est en vente à l'Exposition universelle, section des États-Unis, Galerie des Machines, en papier-carte fort à 1 fr. avec le ciel.

J. A. D.

(Extrait du journal l'Exposition de 1876)

Paris, 15 octobre 1875.
Monsieur le rédacteur en chef,
Comme je vous ai fait découper à découper ce qui peut être utile aux lecteurs de l'Éducation, je tenez à vous



leur connaître la plaque découpée universelle que j'ai exposée dans la partie des machines de la section des États-Unis, au milieu d'une foule de visiteurs.

Cette plaque en carton découpé, d'environ 15 centimètres de longueur sur 8 de largeur, permet de tracer d'une façon correcte et géométrique toutes les lettres de l'alphabet majuscules, minuscules, grecs, hébreux, polonais, les chiffres arabes, l'alphabet grec, et une foule innombrable de dessins.

Tout instituteur pour, à l'aide de cette seule plaque, composer des jeux de lettres solides pour faire les leçons de lecture d'après la méthode naturelle ou intermédiaire. Il peut tracer les lettres de son école de lettres mobiles, pour exposer aux yeux de ses élèves au lieu d'histoire, aux tables de pronoms.

La plaque découpée universelle est si simple qu'un enfant de six ans peut tracer facilement toutes les lettres et chiffres. Elle est indispensable d'enseignement, on peut aussi à grande quantité fabriquer les ornements les plus variés et les plus remarquables.

La plaque découpée universelle de M. J. A. David, est un beau cadeau, et tout acheteur à droit à la clef du système.

Une lettre à M. Chaboussier a été adressée à M. Barthelemy par un journaliste français, M. Victor Naud, qui se félicite de la nouveauté de l'invention de ce merveilleux petit instrument dans tous les établissements d'enseignement public. Pour tout exemple sur les lettres et chiffres, on peut se procurer à l'Exposition, qui se trouve au 1 à acheter la plaque de M. David, quand il se trouve, comme moi, occasion de son existence.

Veuillez agréer, etc.
J. A. D.
(Extrait du journal l'Éducation de 18 octobre, page 676.)

(1) Cette méthode a été exposée dans l'Exposition, par M. Théodore.

EN VENTE

Détail : 26, Passage Jouffroy.

Détail et gros : Chez l'inventeur, 69, avenue du Maine.

Prix, avec clef explicative 1 fr.

Par la poste, franco pour Paris, la France et l'Europe (Union postale) en timbres ou mandats-postes 1 10

NOTA. — Toutes les correspondances doivent être adressées chez l'inventeur, 69, avenue du Maine, Paris.

FORTE REMISE POUR LES ACHATS EN GROS

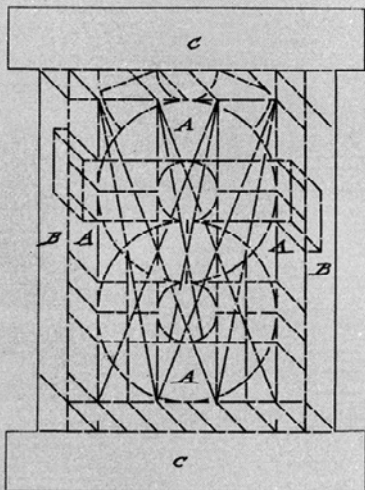
Prière de communiquer ce Prospectus aux personnes auxquelles cette invention pourrait être utile.

Paris. — Typ. MAUVRE et DUBOIS, rue du Cardinal-Lemoine, 41.

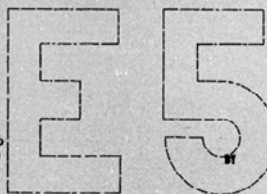
[c7]

Plaque Découpée Universelle, Typography Papers 7,
Reading, 2007

J. A. DAVID.
UNIVERSAL STENCIL-PLATES.
No. 179,686. Patented July 11, 1876.



WITNESSES,
James Davidson
John G. Galt



INVENTOR,
J. A. David
M. M. M. M.
ATTORNEY.

U. S. PATENT OFFICE, WASHINGTON, D. C.

[c8]

Plaque Découpée Universelle, Typography Papers 7,
Reading, 2007



[c9]

Universal, Herbert Bayer, 1925

aāāabbb̄bbc
ddeefḡḡghij
klmnop̄pp
qq̄rr̄fstuv
wxx̄yz
A Ā Ā ck ß

[C10]

Premiers dessins du Futura par Paul Renner (1924-1927),
Futura, une gloire typographique, 2011

[C, 11 — 19]
Description de la lettre



[C11]

Dessin préparatoire de courbes au format Ikarus

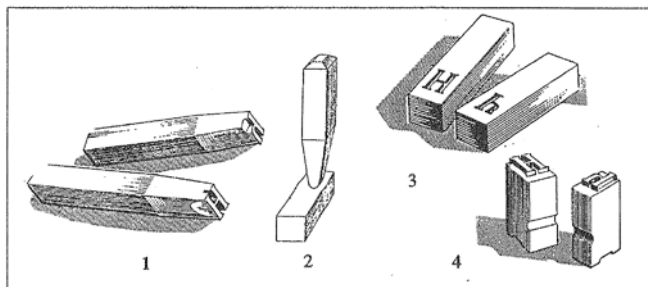
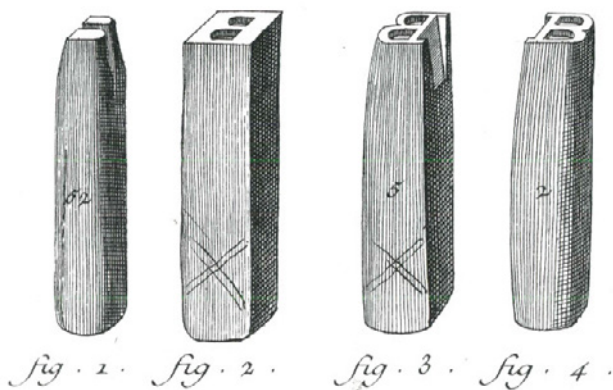


FIG. 2.1 - Principales étapes de la fabrication de caractères en plomb (d'après DREYFUS [7, page 187] et MARSHALL [18, page 44])



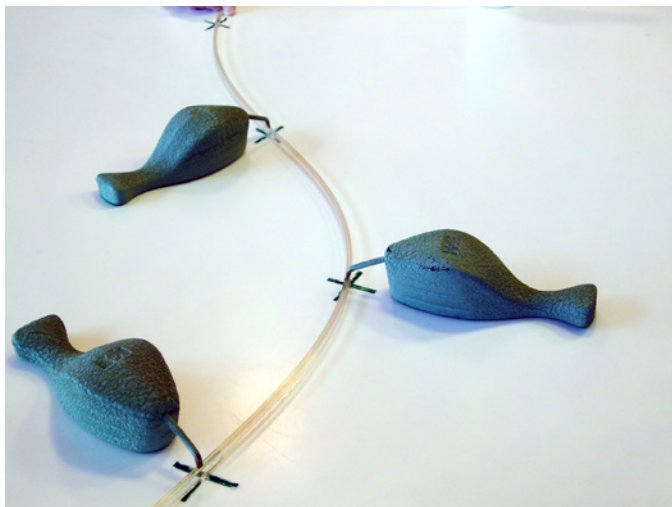
[C12]

La chose imprimée de John Dreyfus,
Encyclopédie de Diderot et d'Alembert



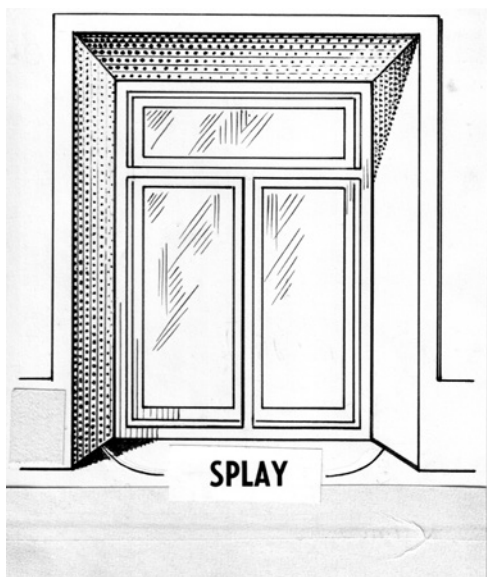
[C13]

Employé de Boeing utilisant des courbes analogiques,
Pearson Scott Foresman

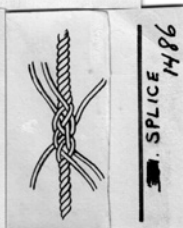
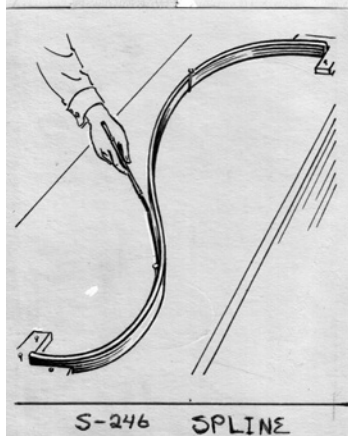


[C14]

Poids de contrôle de courbes analogiques, Pearson Scott Foresman

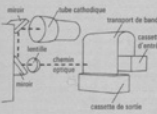

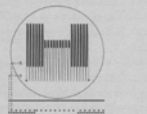



S-244 SPLAY



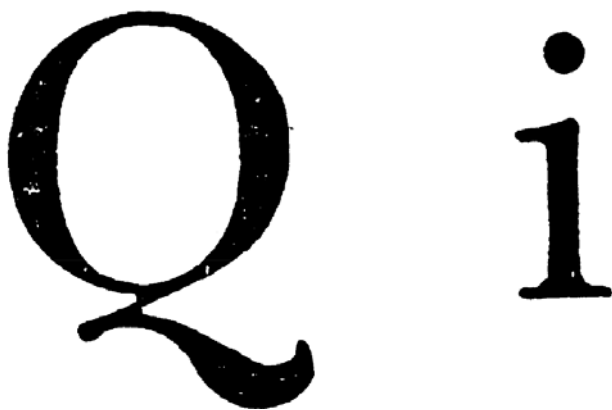
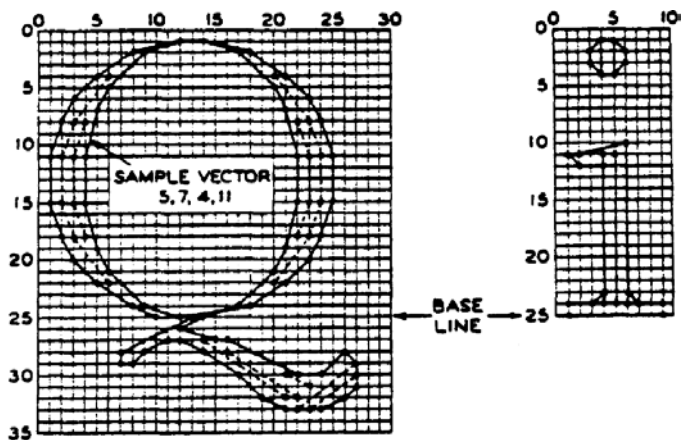
[C15]

Schémas de courbes analogiques, Pearson Scott Foresman

| | | | |
|--|--|---|--|
| <p>/81/ <i>Représentation schématique du chemin optique du rayon cathodique à travers deux miroirs réfléchissant la lumière à 90°.</i></p> | <p>/82/ <i>Caractère numérisé au format Bitmap pour les besoins du flashage et de l'affichage à l'écran.</i></p> | <p>/83/ <i>Représentation schématique de l'enregistrement par le rayon cathodique d'une lettre numérisée.</i></p> | <p>/84/ <i>Description de forme par des lignes de scan verticales (à g.) et description des contours par des vecteurs pour le flashage (à d.).</i></p> |
|  |  |  |  |

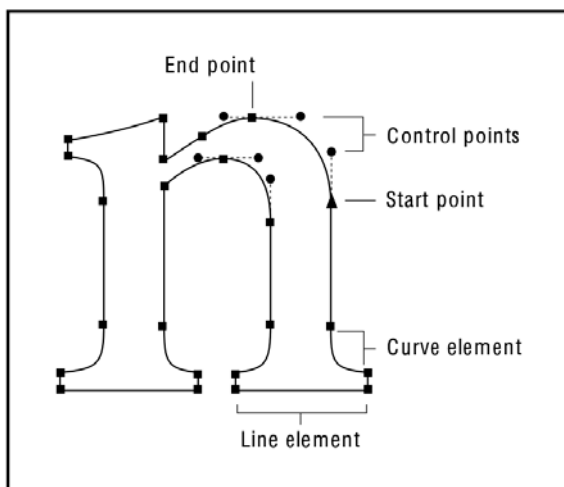
[C16]

Encyclopédie de Diderot, Référence, Année



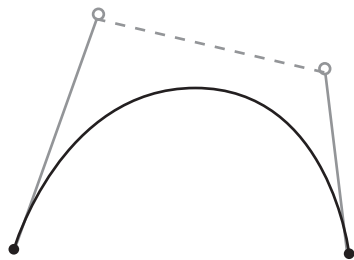
[C17]

Description vectorielle d'un glyphe dans un espace orthonormé,
Peter Karrow [BIBLIOGRAPHIE 3]

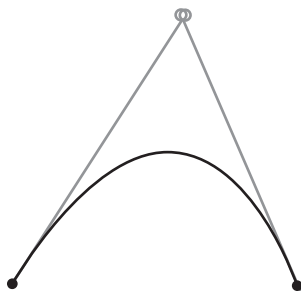


[C18]

Les différents composants descriptifs d'un caractère en courbe de Bézier (cubique), Peter Karrow [BIBLIOGRAPHIE 3]



Courbe cubique

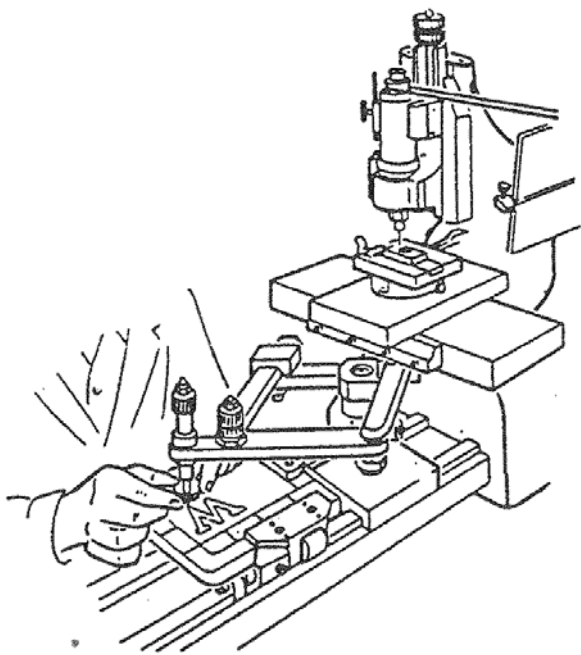


Courbe Quadratique

[C19]

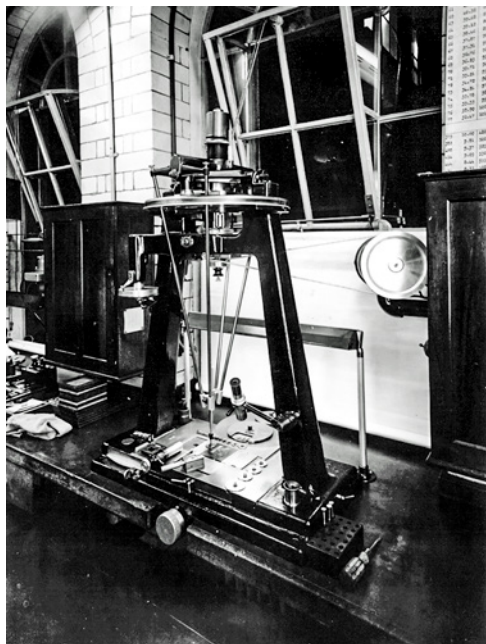
Shémas des courbes cubiques (Postscript) et quadratique (Truetype)

[C, 20 – 31]
Influence formelle de la technique



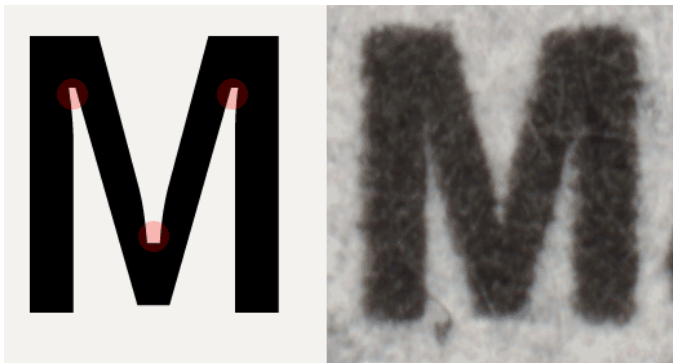
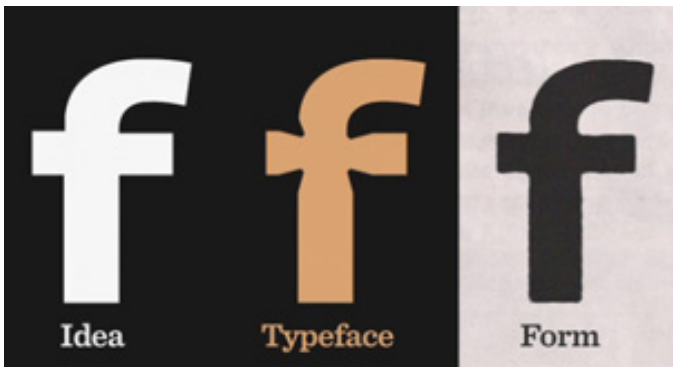
[C20]

Ruptures et continuités dans un changement de système technique,
thèse d'Alan Marshall, Grenoble, 1991



[C21]

Pantographe de Benton

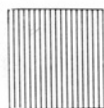


[C22]

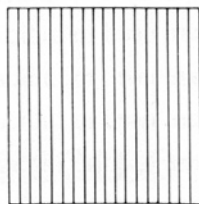
Effet des pièges à encre, caractère de Matthew Carter
(Bell Centennial)



36-POINT EM



36-POINT EM
DIVIDED INTO
18 UNITS



72-POINT EM
DIVIDED INTO
18 UNITS

An 18-unit system.

[C23]

Phototypesetting, A design manual, James Craig

| | | |
|----------|----------|---------|
| | | |
| 18 UNITS | 10 UNITS | 6 UNITS |

Set-width, or unit value, for three characters.

| CHARACTERS | SET-WIDTH |
|--|-----------|
| i, j, l | 4 |
| f, t, l, ., , | 5 |
| r | 6 |
| c, k, s, v, x, y, z, J | 9 |
| a, b, d, e, g, h, n, o, p, q, u, L 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 | 10 |
| F, T, Z | 11 |
| A, B, E, K, P, V, X, Y | 12 |
| w, C, D, H, N, U, R | 13 |
| G, O, Q | 14 |
| m, M | 15 |
| W | 17 |

*Set-width, expressed in units, for 12-point
Helvetica Roman.*

[C24]

Phototypesetting, A design manual, James Craig



[C25]

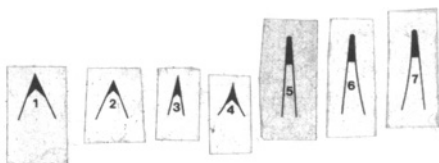
Marques de contraintes du dessin pour photocomposition,
A. Frutiger [BIBLIOGRAPHIE 12]



FIG. 2.6 - Les caractères doivent être redessinés en fonction de la technologie employée: en A, le caractère théorique; en B ce qu'il donnerait en photocomposition; en C, la façon dont il doit être redessiné pour qu'une photocomposeuse donne le dessin A; en D, ce que donnerait une machine à écrire à partir de A; en E le caractère redessiné pour qu'une machine à écrire donne le caractère A (d'après FRUTIGER [33]).

[C26]

Différents effets possibles sur un même dessin d'un flashage à la photocomposition, *Typography with the IMB Composer*, A. Frutiger



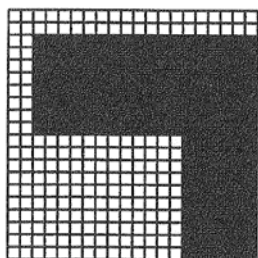
[C27]

Correction optique d'un caractère avant flashage
à la photocomposition, A. Frutiger [BIBLIOGRAPHIE 12]



[C28]

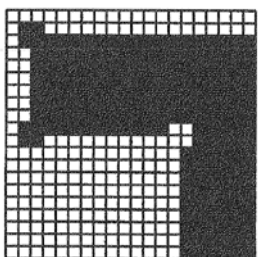
Différents types de tramage d'un dessin vectoriel,
A. Frutiger [BIBLIOGRAPHIE 12]



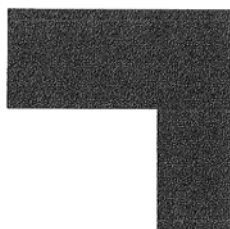
a



b



c

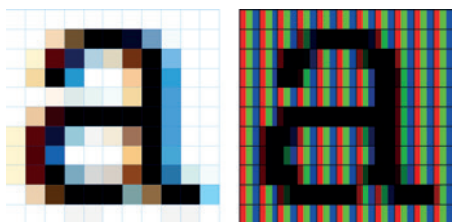
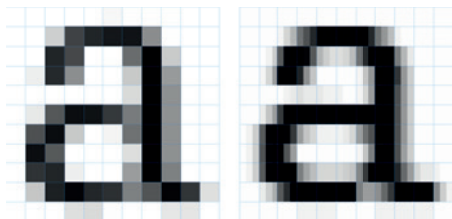
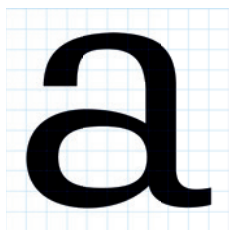


d

FIG. 3.7 - Si on donne le plan de bits (a), on obtient l'image (b); pour avoir l'image (d), il faut tricher et donner le plan (c). Voir aussi la figure 2.6. D'après RUBINSTEIN [115, pages 79-80]

[C29]

Correction optique sur trame de pixels, Digital Typography,
an introduction to type and composition, Richard Rubinstein, 1988



Resolution
Resolution

Resolution
Resolution

[C30]

Diiférents types de rendus sous pixel, Benton Modern

Authority may review records
Captain Ronald Simons started out
CANDIDATE SPOKE AT MEETING
SAILORS REPORTED CALM WEATHER
Many ideas will not be known
Human nature has made headway
ALPINE CLIMBERS SCALE PEAKS
MILLIONAIRE BRIDE PLANS CRUISE

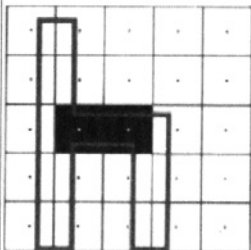
Noted foreign artists exposed
Many problems thought trifling
BOOKS OF FICTION SELDOM FAIL
Enormous quantities of paper
Modern printers recognize fact
SOLDIER UNVEILED MONUMENT
Western city amazed visitors
Exceptional planetary activity

[C31]

Swift (1985) et Gulliver (1993), Gerrard Unger,

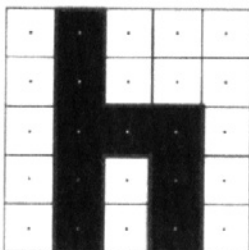
[C, 32 — 35]
Le Hinting

grid fit module off

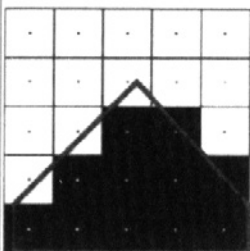


arbitrary stem positions
in double fine Target Grid

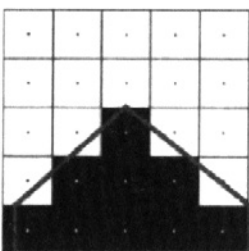
grid fit module on



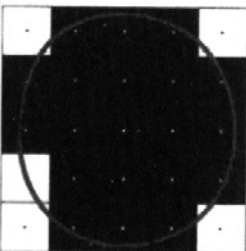
stem positions are rounded
exactly on Target Grid lines



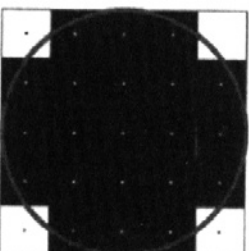
blunt peaks can occur



sharp peak due to special peak
positioning



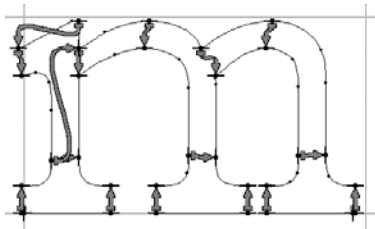
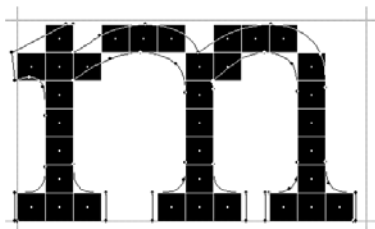
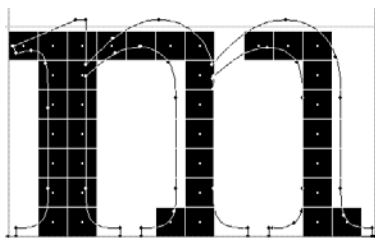
arbitrary curve extremes positions
can lead to unwanted pimples



curve extremes are rounded exactly
on Target Grid lines as stem lines

[C32]

Principes de correction du tralage par le Hinting,
Peter Karrow [BIBLIOGRAPHIE 3]



[C33]

Exemple de Hinting TTF avec le logiciel
Visual Open Layout Tool (VOLT) de Microsoft

e e s

e e e

e e e

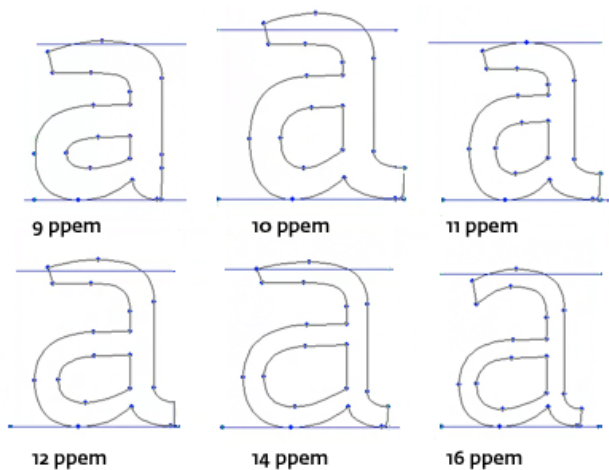
e e s

e e e

e e e

[C34]

Exemple de correction Hinting (Avant / Après)



[C35]

Effet du hinting sur l'emplacement des points
lors d'un changement de corps, Typothèque

[D, 1 — 6]

Les outils du créateur typographique
Outils analogiques

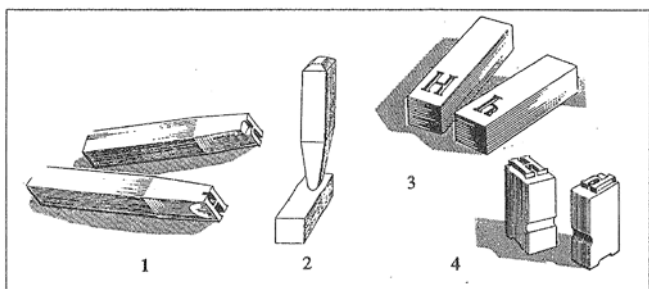
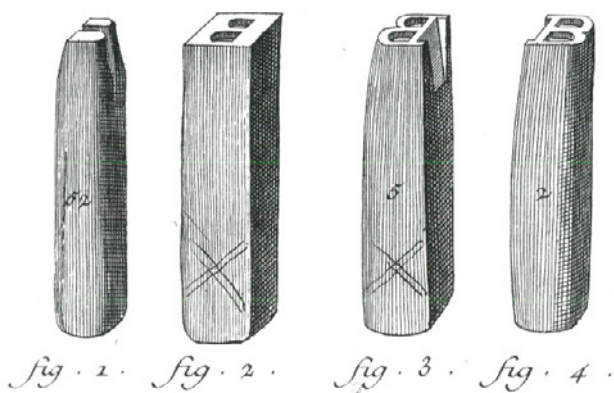
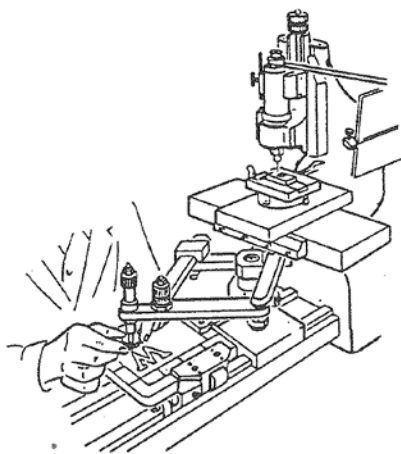
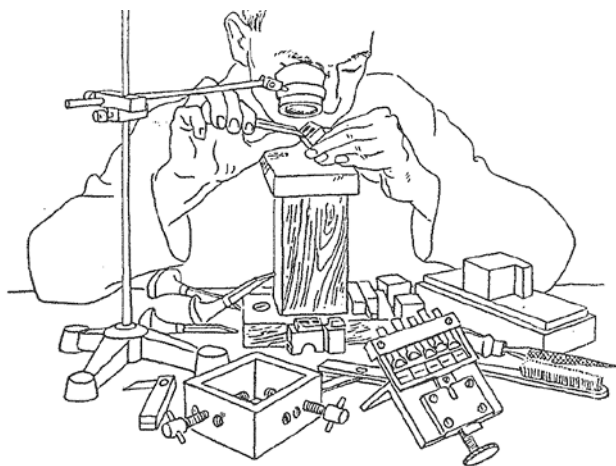


FIG. 2.1 - Principales étapes de la fabrication de caractères en plomb (d'après DREYFUS [7, page187] et MARSHALL [18, page 44])



[D1] [C13]

Encyclopédie de Diderot et d'Alembert,
La chose imprimée de John Dreyfus



[D2]

Letter of credit: a view of type design, Walter Tracy, Londres, 1986



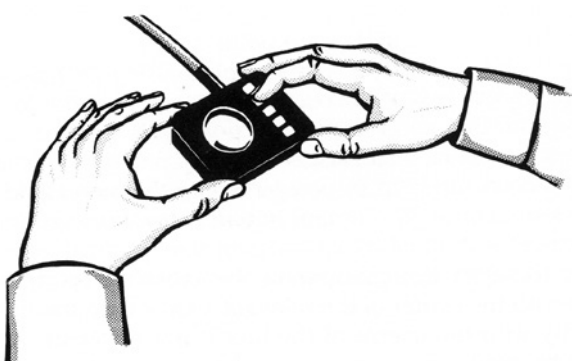
[D3]

IBM, 1965



[D4]

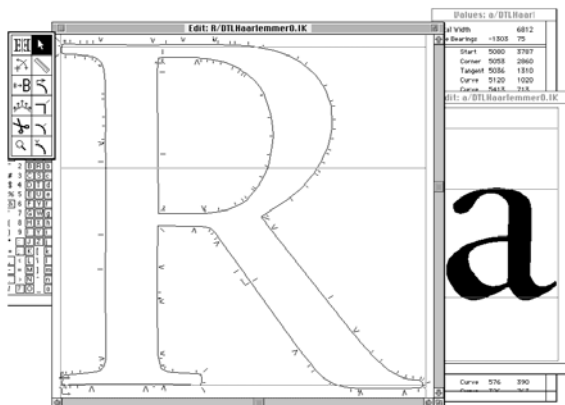
Macintosh 128k, Wikimedia



[D5]

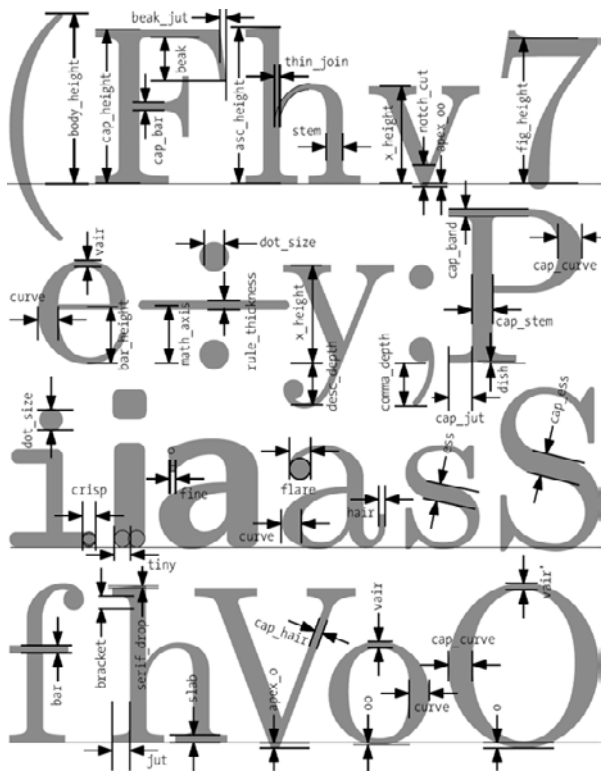
Ikarus mouse

[D, 6 — 17]
Outils numériques



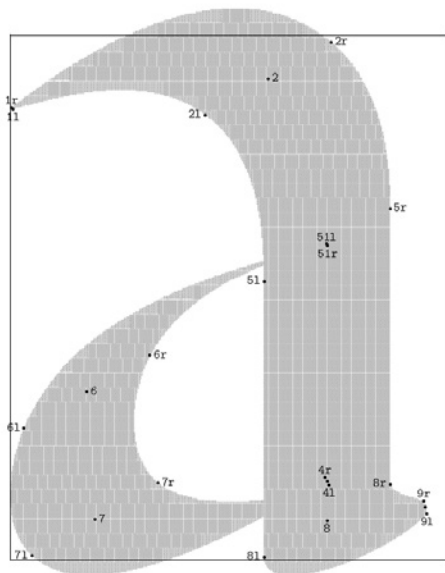
[D6]

Capture d'écran, Logiciel de création de caractères Ikarus sur Mac



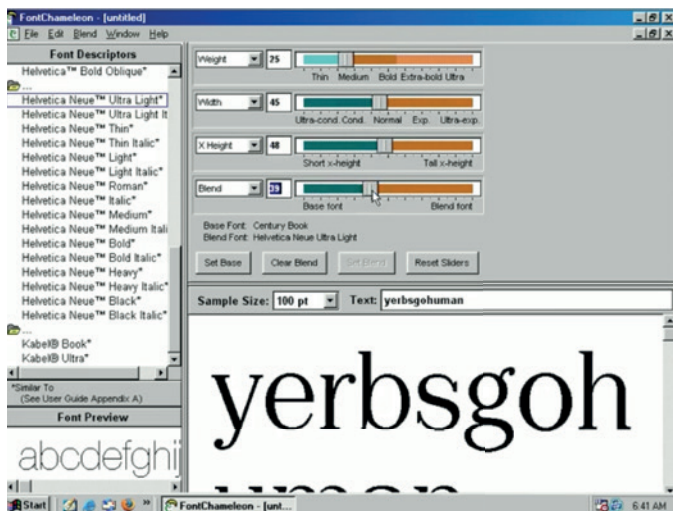
[D7]

Glyph metafont et ses points descriptifs,
 Donald Knuth Metafont book, extrait de Computer Modern



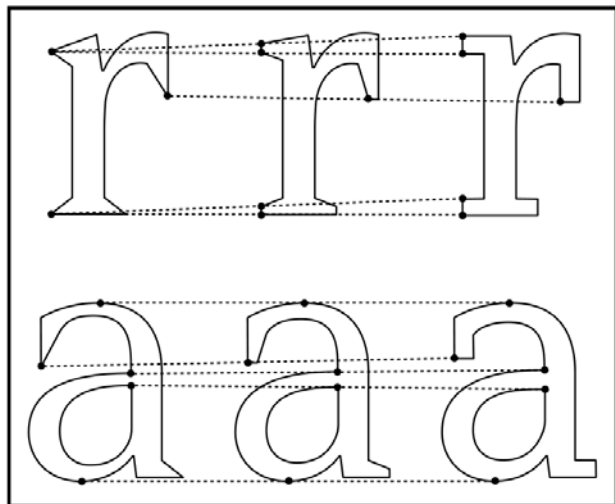
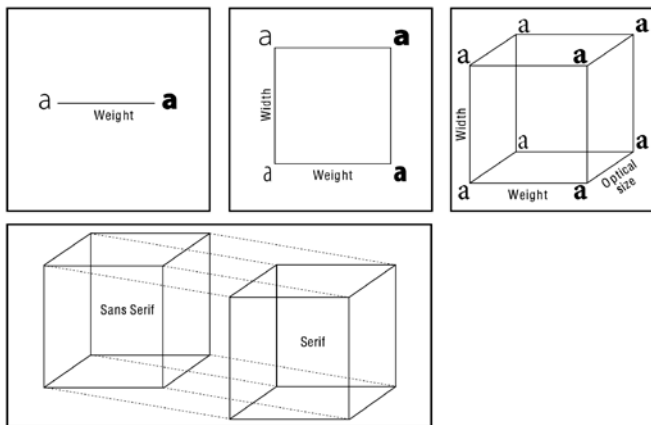
[D8]

Glyph metafont et ses points descriptifs,
 Donald Knuth Metafont book, extrait de Computer Modern



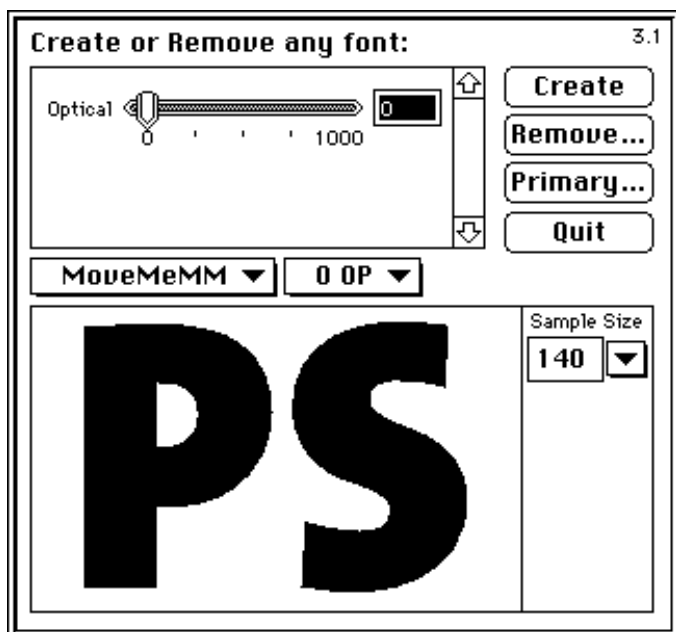
[D9]

Logiciel Font Chameleon sur Windows,
FontLab, 1994



[D10]

Designing Multiple Master Typefaces,
1995, Adobe Multiple Master font format



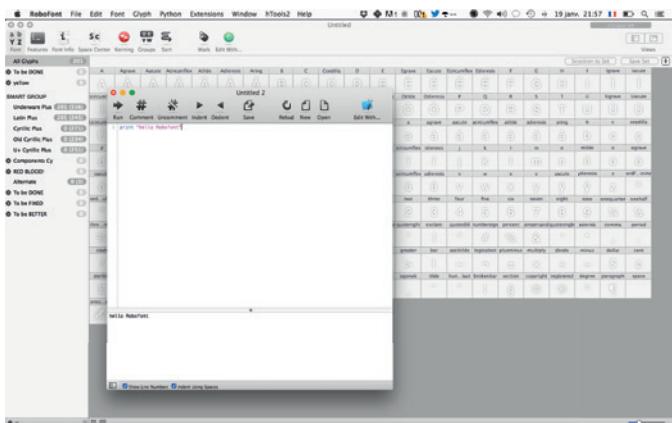
[D11]

Capture d'écran, Interface d'Adobe Type Manager
et d'une instance Multiple Master Font



[D12]

Capture d'une boîte de dialogue Quickdraw GX



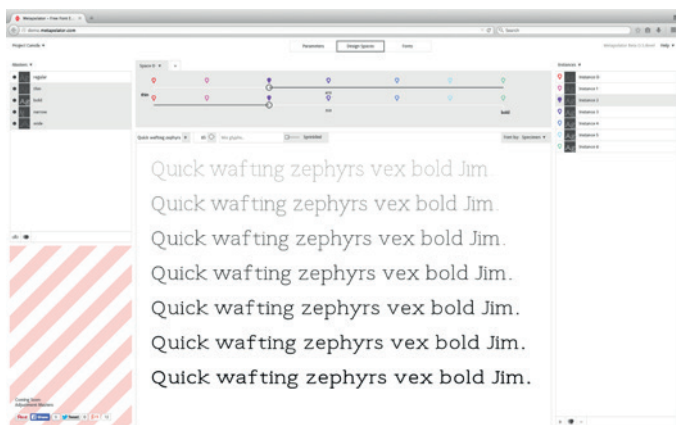
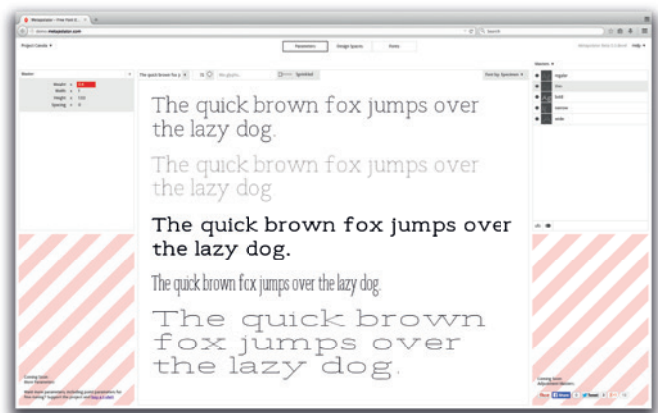
[D13]

Capture d'écran de l'interface de RoboFont,
fenêtre de scripting Python, Frédéric Berlaen



[D14]

Capture d'écran de Superpolator 3, logiciel d'interpolation,
Erick Van Blockland



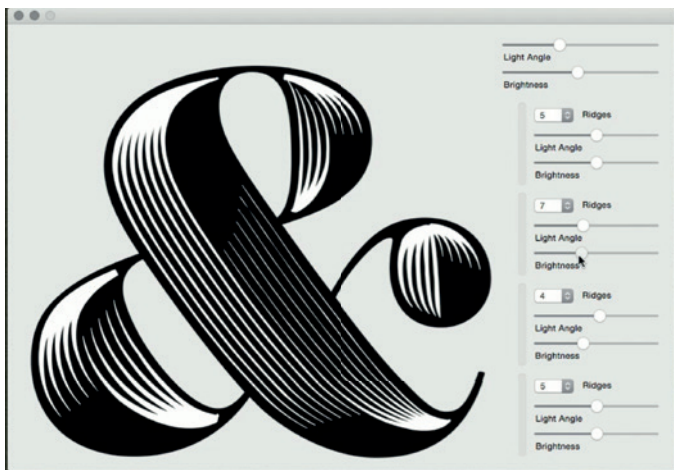
[D15]

Capture d'écran, Démo Metapolator, metapolator.com



[D16]

Capture d'écran, prototipo.io

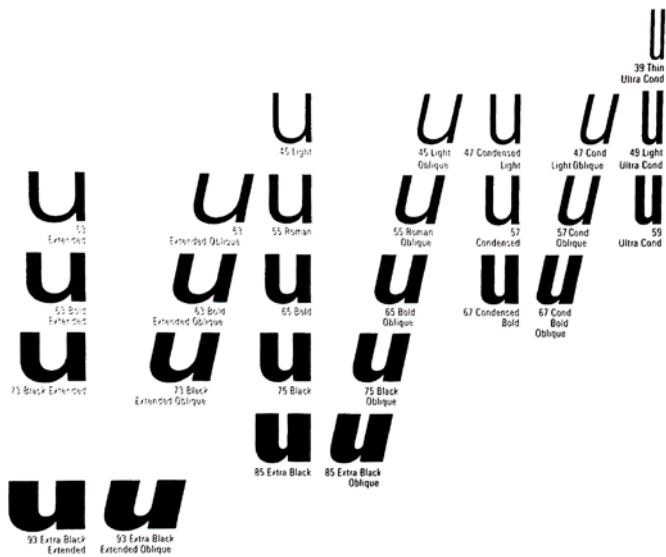


[D17]

nterface logiciel utilisée pour générer la fonte Obsidian,
Andy Climer, Vimeo, 2014

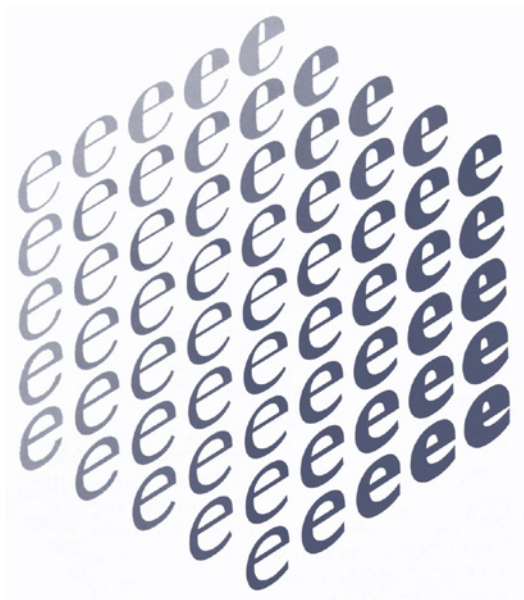
[E, 1 — 4]

Famille de caractère
Famille étendue statique



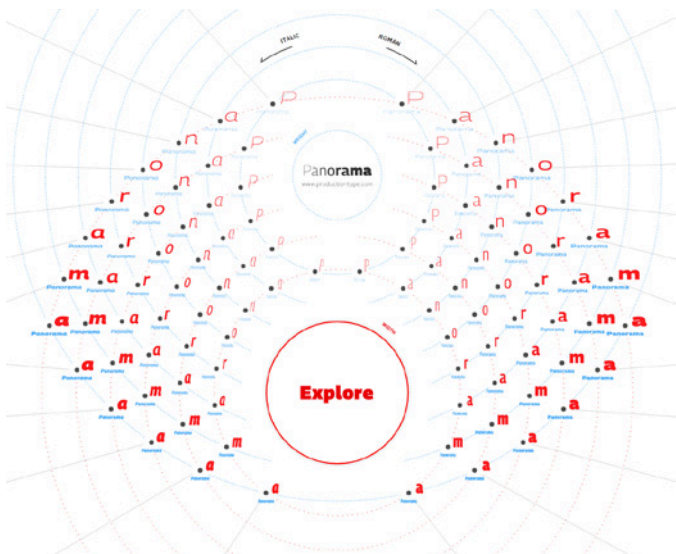
[E1]

Spécimen Univers, A. Frutiger, 1957



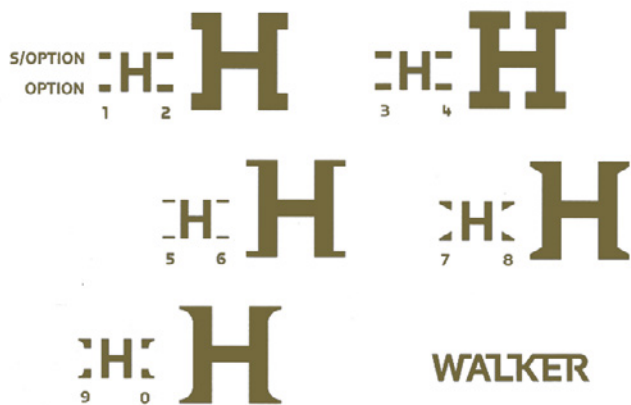
[E2]

Gerrit Noordzij cube, The stroke of the pen, 1982



[E3]

Spécimen ligne de la super famille Panorama, Jean Baptiste Levée,
Spécimen de la super famille Thesis, Lucas de Groot



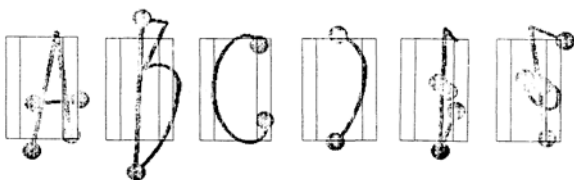
Decorativa, 1994

[E4]

Walking Art center, Matthew Carter, 1990

History, Typotheque, Peter Bilak, 2010

[E, 5 — 11]
Fontes dynamiques



(I refuse to show you the first form of the letter 'G'.) But by 4 p.m. I was ready to make the first trial setting of text:

THIS 'PUNK' ALPHABET, INSPIRED BY MARJAN'S LECTURE LAST NIGHT, WAS DESIGNED BY METAFONT'S RANDOM NUMBER GENERATOR.

THE QUICK BROWN FOX JUMPED OVER THE LAZY HAMBURGER FONTS.

[E5]

Punk, Donald Knuth, Tug boat n°2, 1988

L'ORAGE

CE POINT
DE LA TERRE ET L'ÉCORCE.
DÙ LE TRONC.

LE SPIR.
LE MATIN.
DES PLUIES
SUR LES NAVIRES.

LE PAS
DU CHEVAL.
LE TON.
DES SABOTS.
SUR LE SOL.
EST MAT

LE THÉ.
TRÈS CHAUD.
QUAND IL
EST
FORT.
ET ÂCRE.

L'EAU.
NOÏRE.
SUR UN TAPIS
DE FEUILLES.
ROUSSÉS.

L'ORAGE

CE POINT
DE LA TERRE ET L'ÉCORCE.
DÙ LE TRONC.

LE SPIR.
LE MATIN.
DES PLUIES
SUR LES NAVIRES.

LE PAS
DU CHEVAL.
LE TON.
DES SABOTS.
SUR LE SOL
EST MAT.

LE THÉ.
TRÈS CHAUD.
QUAND IL
EST.
FORT.
ET ÂCRE.

L'EAU.
NOÏRE.
SUR UN TAPIS
DE FEUILLES.
ROUSSÉS.

[E6]

Le Delorme, un caractère modulaire et dépendant du contexte,
Jacques André, 1990

The Desperate **Chainsaws**
19: Open Mike Night* (A)
20: NoMoreCurves (UK)
+ Bassment (Amsterdam)
21: Urban Death Ray (D)
Assinine Megaturbidites
22. Sedimäntary Bodeez
23. De@thSchr!ft (USA)
LastPartay at this venue!
>**PLZ**<Add2Shoppingcart?
And Tonight Only: Dädadaist Pønks

[E7]

Beowolf, Letterror, 1995

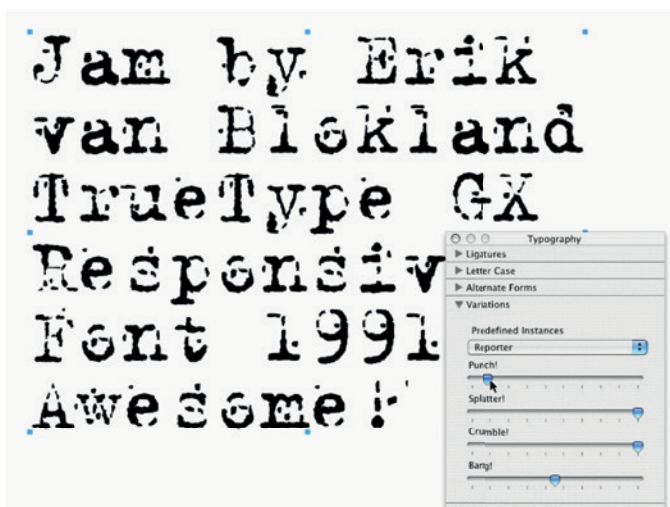
THE QUICK BROWN F

THE QUICK BROW

THE QUICK BRO

[E8]

Lance Hidy, Adobe Penumbra, Multiple Master Font, 1994



[E9]

Jam, Erik Van Blokland, TTGX, 1991



[E10]

Utilisation du Skia par Philippe Millot, Édition Cent Pages

Sitka

A new family of 24 size-specific fonts designed by Matthew Carter. Built by Tiro Typeworks Ltd. Available with Microsoft Windows 8.1

Banner #6pt

Tunguska

Display #6pt

Enormously

Heading #9pt

powerful airburst

explosion in Siberian wilderness, June 1908

Text #8pt

At around 07:17 local time, Evensk natives and Russian settlers in the hills northwest of Lake Balkhal observed a column of bluish light, nearly as bright as the Sun, moving across the sky. About 10 minutes later, there was a flash and a sound similar to artillery fire. Eyewitnesses closer to the explosion reported the sound source moving east to north.

Small #6pt

The events were accompanied by a shock wave that knocked people off their feet and broke windows hundreds of kilometers away. The majority of witnesses reported that the sound had the character, and the lighting of the explosion. Eyewitness accounts differ as to the sequence of events and their overall direction. The explosion registered on seismic stations across Eurasia. In some places the shock wave would have been registered as an earthquake of 1.0 on the Richter scale. It penetrated the thickness of sedimentary pressure strata enough to be detected in Great Britain. Over the next few days, night skies in Asia and Europe were again. It has been theorized that this was due to light passing through high altitude ice particles formed at secondary low temperature. Observations that occurred days, when giant auroras in several areas throughout the Northern Hemisphere. The Smithsonian Astrophysical Observatory and the Mount Wilson Observatory observed a decrease in atmospheric transparency that lasted for several months, from unspecified date.

Banner Bold #6pt

Tunguska

Display Bold #6pt

Enormously

Heading Bold #9pt

powerful airburst

explosion in Siberian wilderness, June 1908

Text Bold #8pt

At around 07:17 local time, Evensk natives and Russian settlers in the hills northwest of Lake Balkhal observed a column of bluish light, nearly as bright as the Sun, moving across the sky. About 10 minutes later, there was a flash and a sound similar to artillery fire. Eyewitnesses closer to the explosion reported the sound source moving east to north.

Small Bold #6pt

The events were accompanied by a shock wave that knocked people off their feet and broke windows hundreds of kilometers away. The majority of witnesses reported that the sound had the character, and the lighting of the explosion. Eyewitness accounts differ as to the sequence of events and their overall direction. The explosion registered on seismic stations across Eurasia. In some places the shock wave would have been registered as an earthquake of 1.0 on the Richter scale. It also penetrated the thickness of sedimentary pressure strata enough to be detected in Great Britain. Over the next few days, night skies in Asia and Europe were again. It has been theorized that this was due to light passing through high altitude ice particles formed at secondary low temperature. Observations that occurred days, when giant auroras in several areas throughout the Northern Hemisphere. The Smithsonian Astrophysical Observatory and the Mount Wilson Observatory observed a decrease in atmospheric transparency that lasted for several months, from unspecified date.

Banner Italic #6pt

Tunguska

Display Italic #6pt

Enormously

Heading Italic #9pt

powerful airburst

explosion in Siberian wilderness, June 1908

Text Italic #8pt

At around 07:17 local time, Evensk natives and Russian settlers in the hills northwest of Lake Balkhal observed a column of bluish light, nearly as bright as the Sun, moving across the sky. About 10 minutes later, there was a flash and a sound similar to artillery fire. Eyewitnesses closer to the explosion reported the sound source moving east to north.

Small Italic #6pt

The events were accompanied by a shock wave that knocked people off their feet and broke windows hundreds of kilometers away. The majority of witnesses reported that the sound had the character, and the lighting of the explosion. Eyewitness accounts differ as to the sequence of events and their overall direction. The explosion registered on seismic stations across Eurasia. In some places the shock wave would have been registered as an earthquake of 1.0 on the Richter scale. It also penetrated the thickness of sedimentary pressure strata enough to be detected in Great Britain. Over the next few days, night skies in Asia and Europe were again. It has been theorized that this was due to light passing through high altitude ice particles formed at secondary low temperature. Observations that occurred days, when giant auroras in several areas throughout the Northern Hemisphere. The Smithsonian Astrophysical Observatory and the Mount Wilson Observatory observed a decrease in atmospheric transparency that lasted for several months, from unspecified date.

Banner Bold Italic #6pt

Tunguska

Display Bold Italic #6pt

Enormously

Heading Bold Italic #9pt

powerful airburst

explosion in Siberian wilderness, June 1908

Text Bold Italic #8pt

At around 07:17 local time, Evensk natives and Russian settlers in the hills northwest of Lake Balkhal observed a column of bluish light, nearly as bright as the Sun, moving across the sky. About 10 minutes later, there was a flash and a sound similar to artillery fire. Eyewitnesses closer to the explosion reported the sound source moving east to north.

Small Bold Italic #6pt

The events were accompanied by a shock wave that knocked people off their feet and broke windows hundreds of kilometers away. The majority of witnesses reported that the sound had the character, and the lighting of the explosion. Eyewitness accounts differ as to the sequence of events and their overall direction. The explosion registered on seismic stations across Eurasia. In some places the shock wave would have been registered as an earthquake of 1.0 on the Richter scale. It also penetrated the thickness of sedimentary pressure strata enough to be detected in Great Britain. Over the next few days, night skies in Asia and Europe were again. It has been theorized that this was due to light passing through high altitude ice particles formed at secondary low temperature. Observations that occurred days, when giant auroras in several areas throughout the Northern Hemisphere. The Smithsonian Astrophysical Observatory and the Mount Wilson Observatory observed a decrease in atmospheric transparency that lasted for several months, from unspecified date.

[E11]

Sitka par Matthew Carter et Tiro Typeworks, 2013

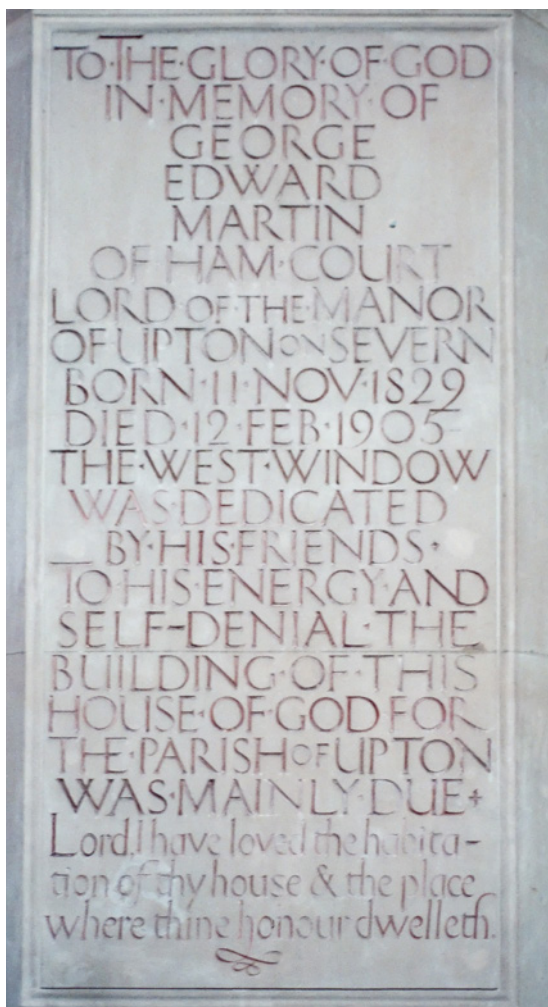
[F, 1 — 12]

Relation typographie/contexte
Contrainte de l'espace



[E1]

Inscription lapidaire



[E2]

Gravure Lapidaire, auteur et date inconnus



[E3]

Tag, auteur et date inconnus



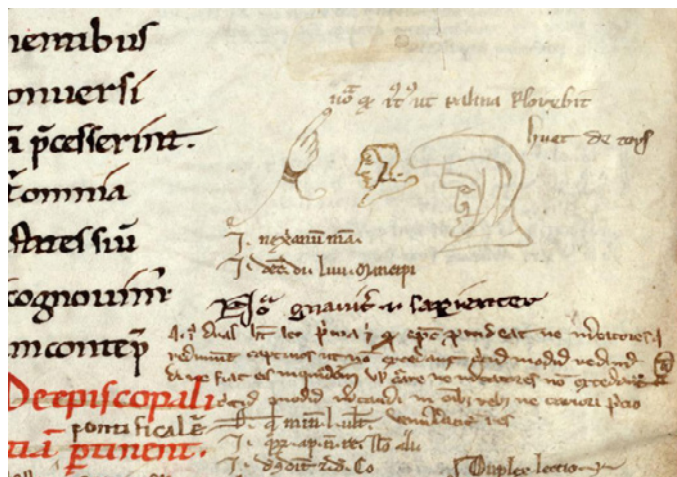
[E4]

Gravure Lapidaire funraire, auteur et date inconnus



[E5]

Extrait de Typography Paper 7, citation d'Eric Gill



[E6]

Glose dans un texte du Moyen Âge, date et auteur inconnus



[E7]

Affiche du XIX^e siècle



[E8]

Affiche publicitaire pour la marque de peinture Ripolin, 1913

A. VOKALE:
 AA E J O O O O U U:
 AA A A O O O O U U U U
 AA A A O O O O U U U U
 AA A A O O O O U U U U

LANG
 KURZ

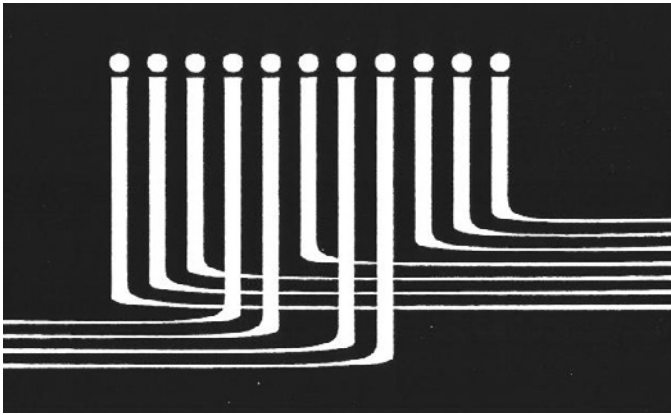
B. KONSONANTEN:

| | | | |
|---|----|-----|------|
| | G | d | b |
| | K | T | P |
| | J | -S | W |
| h | dh | sch | -S F |
| | NG | | N M |
| | R | L | |

| | | | | |
|--|---|---|---|--|
| | J | I | | |
| | I | T | | |
| | | f | t | |
| | f | f | f | |
| | J | | f | |
| | f | t | f | |

[E9]

Écriture systématique, K. Schwitters, 1927



the room
the room

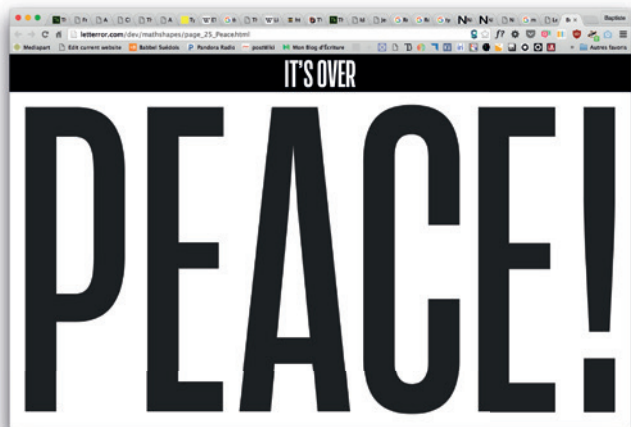
[E10]

Exemple de justification sémitique, contre une justification
par le blanc, Jacques André [BIBLIOGRAPHIE 8]

| SHUFFLEBOARD STANDINGS | | |
|------------------------------------|--------------------------------|------------------------------------|
| PILOT DIVISION | CHERRY DIVISION | HAMMER DIVISION |
| FUSSY PUCKERS 5 1 | SHUFFLEUPAGUS 6 0 | FLOUR, WATER, BISCUIT, EAT IT! 6 0 |
| GOWANUS DOLPHINS 5 1 | SHUFFLEBAGS 5 1 | ATOMIC FIREBALLS 5 1 |
| BEGIN FOR MORE 4 2 | SWIFT BISCUITS 5 1 | PERPENDICULAR BISECTORS 5 1 |
| BROOKLYN NETS OF SHUFFLEBOARD 4 2 | TRUFFLE SHUFFLE 5 1 | WILD TANG 5 1 |
| BROOKLYN SHUFFLE 4 2 | AINT NOthin' BUT A G TANG 4 2 | DINO SLIDERS 4 2 |
| GLORIA AGNES SAUL & MORTY 4 2 | POOTIE TANG 4 2 | THE FLAMINGO KIDS 4 2 |
| G O N E R 4 2 | SHUFFLEBIRDS 4 2 | G-SQUARED 4 2 |
| KITCHENSURFING 4 2 | WHEN PUSH COMES TO SHUFFLE 4 2 | SUPERTANG 4 2 |
| SUICIDE ALLEY 4 2 | THE MEATBALLERS 3 3 | BISCATZ & CROSSBONES 3 3 |
| ARECIBO 5 MINUTES 3 3 | NO SLEEP 'TIL BOCA 3 3 | THE JERKS 3 3 |
| BOARD TO DEATH 3 3 | SCIENCE OVEN 3 3 | THE MIGHTY BISCUITS 3 3 |
| GOWANUS SWIM CLUB 3 3 | SHIFTY MOTHER PUCKERS 3 3 | PSYCHOTIC SHUFFLERS 3 3 |
| HELBRAUN LEVEY & O'SHUFFLEBALE 3 3 | BISCUITCASES 2 4 | RISKY BISCUITS 3 3 |

[E11]

Font-to-width, Nick Sherman, 2014

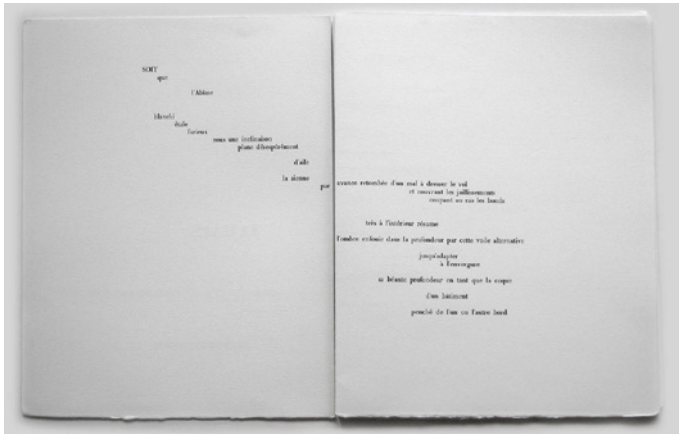


[E12]

Digital Lettering, Letterror, 2015

[E, 13— 15]

Relation forme/sens



[E13]

Un coup de dés n'abolira jamais le hasard, S. Mallarmé, 1897



[E14]

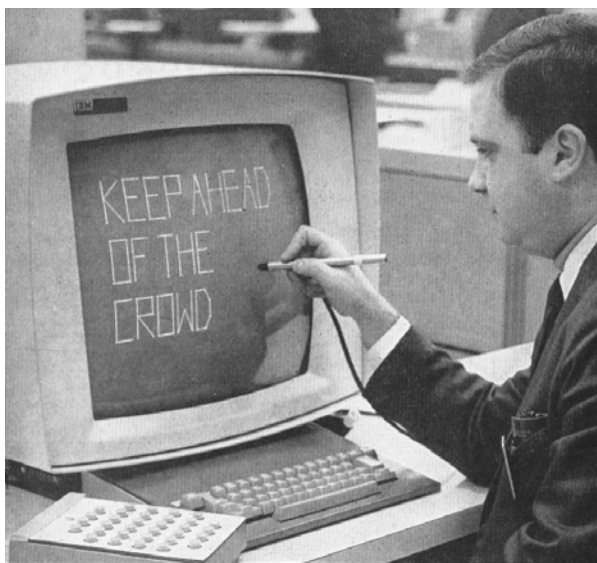
La cantatrice chauve, Massin, Gallimard, 1964

graphic graphic graphic ~~graphic~~
light **black heavy** square
short extended **one** round
contrasted

frack

[E15]

Benjamin Gomez, Dépli design, 2004



Imprimé à l'ESAD Valence le 30/02/2016 sur Olin
Bulk Gold 175, 80g et Maine Gloss Certifié Perf
135g. Texte composé en Lyon Text de Kai Bernau et
Neue Haas Unica, design original de Team '77 et
revival de Toshi Omagari pour Monotype.

