

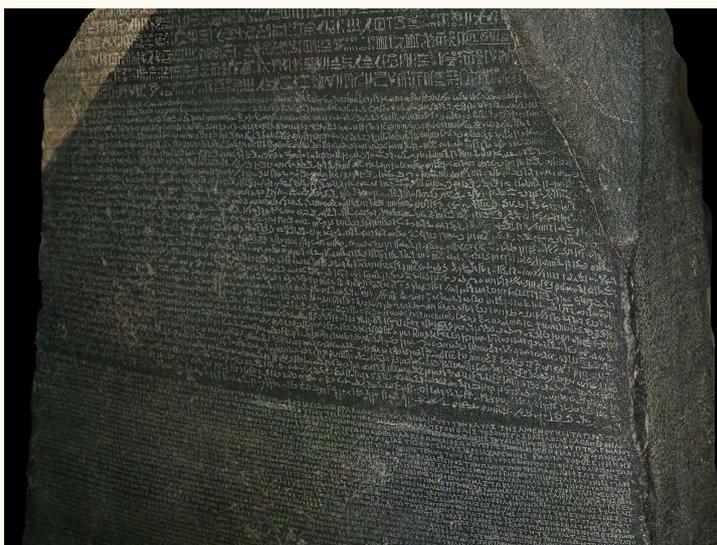
🌀 EN ROUTE VERS NOTO

Note de la rédaction du *TUGboat* : ce texte est la transcription de l'exposé de Steven Matteson lors de la conférence *TUG 2020*. Elle n'a fait l'objet que de retouches mineures. Certaines de ses illustrations sont ici omises. Leur intégralité, ainsi qu'une vidéo de la conférence, sont disponibles en ligne à l'adresse de la conférence : <https://tug.org/tug2020>.

La famille de polices Noto est l'une des plus grandes entreprises de l'histoire de la création de caractères. Elle n'alla certainement pas en ligne droite, d'un point A à un point B. J'ai été impliqué de façon intermittente pendant 14 ans, et environ 60 autres personnes y ont contribué jusqu'à ce jour. Cela n'inclut pas les efforts déployés pour les polices chinoises, japonaises et coréennes, dont les gens d'Adobe pourraient vous parler.

Pour les besoins de cet exposé, la route qui nous mène vers Noto commence par la pierre de Rosette (figure 9) : un fragment de tablette large d'environ un mètre, semblable au granit, avec un travail remarquable de la main et du ciseau, épelant soigneusement un décret impérial dans trois systèmes d'écriture différents.

FIGURE 9 – La pierre de Rosette, 196 avant notre ère

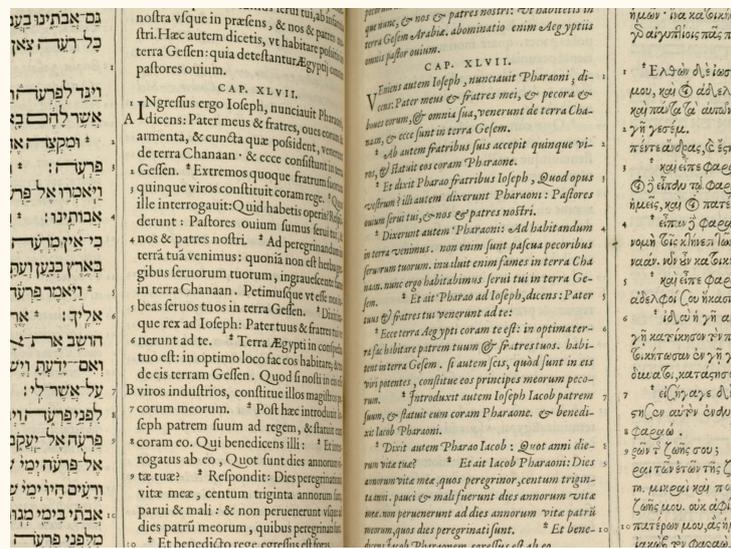


L'importance historique de cette pierre est légendaire. Je me souviens très bien de l'avoir étudiée en cours d'histoire à l'âge de 12-13 ans. Nous devions fabriquer notre propre version en pâte à modeler et y inscrire nos propres messages à l'aide d'un cure-dent. Nous avons même organisé un concours pour essayer de traduire les messages des autres. Merci aux écoles publiques d'Oak Park, dans l'Illinois !

Il s'agit du « décret de Memphis » du roi Ptolémée V, promulgué en 196 avant notre ère, au cours d'une période de turbulences politiques et de bouleversements culturels. La pierre de Rosette est un exemple fascinant des efforts laborieux déployés pour produire un document en plusieurs langues — en l'occurrence en hiéroglyphes, en écriture démotique et en grec ancien — dans le but de permettre à plusieurs cultures et générations de comprendre ce message unique.

Avance rapide jusqu'en 1573, où la Bible polyglotte de Christophe Plantin, financée par le roi Philippe II d'Espagne, est imprimée à Anvers (figure 10).

FIGURE 10 – La Bible polyglotte de Christophe Plantin, 1573



Les Écritures y sont imprimées en six volumes ; un dictionnaire multilingue en deux volumes les accompagnaient. Le texte est traduit en hébreu, en grec, en araméen, en syriaque et en latin. La typographie est stupéfiante de beauté, de simplicité et de minutie.

Le détail d'une double page présenté ci-dessus montre l'écriture hébraïque à l'extrême gauche et le latin dans une police de caractères romaine droite. À l'extrême droite se trouve un beau texte grec d'aspect cursif et fluide ; le latin qui l'accompagne est en italique pour compléter l'aspect et la texture du grec. Ce détail typographique contribue à unifier et à harmoniser la page malgré les différences entre les systèmes d'écriture multilingues.

Pour moi, la complexité du formatage et la qualité de l'impression sont d'une beauté inconcevable.

À partir de 1654, environ 70 ans plus tard, en Angleterre, l'évêque Brian Walton a commencé à travailler sur sa bible polyglotte (figure 11 page ci-contre). Il publie neuf traductions : araméen, hébreu, syriaque, arabe, samaritain, éthiopien, grec et latin. Cette production est financée par souscription et non par une subvention d'un gouvernement ou d'un organisme ecclésiastique. En l'espace d'un an, l'évêque a trouvé 400 contributeurs privés qui attendaient le livre fini.

Certains la considèrent comme la moins belle de toutes les bibles polyglottes. Cela peut s'expliquer par le fait qu'il ne s'agissait pas d'un projet financé par la royauté et assorti d'un faste royal proportionnel. Ou peut-être à cause de la complexité des neuf traductions par rapport aux quatre ou cinq traductions des ouvrages précédents, ce qui met le typographe au défi d'assembler une page harmonisée.

FIGURE 11 – La bible polyglotte de l'évêque Brian Walton, 1654

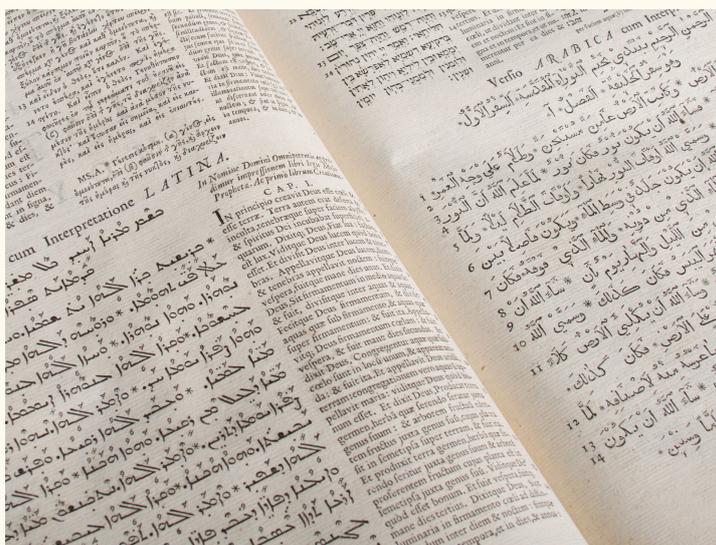
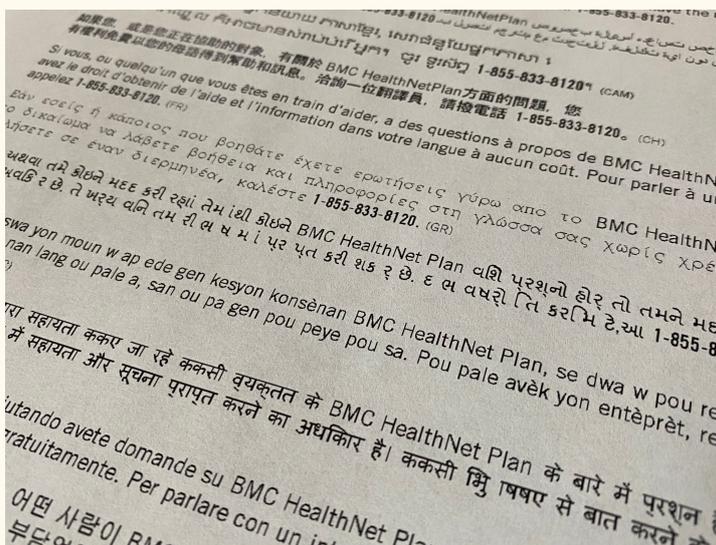


FIGURE 12 – Une page polyglotte contemporaine courante



Mais pour moi, la réalisation de Walton, la complexité du formatage et la qualité de l'impression sont d'une beauté inconcevable. Surtout si l'on la compare avec la situation dans laquelle nous nous trouvons 400 ans plus tard : il n'est qu'à voir ce texte détaillant les prestations d'une compagnie d'assurance quelconque (figure 12).

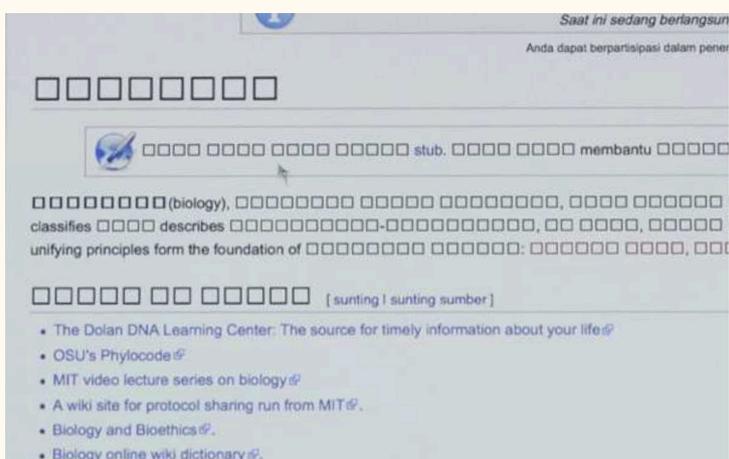
Cette page polyglotte est aujourd'hui courante et facile à réaliser avec nos outils de composition actuels. Cependant, comme de nombreux formulaires génériques ou institutionnels que nous voyons tous les jours, cette page peut être grandement améliorée. La longueur des lignes est excessive pour la plupart des textes représentés ici. La variété des styles de caractères donne l'impression qu'il s'agit d'une demande de rançon plutôt que d'un document sérieux. L'audace de certaines traductions les fait paraître beaucoup plus importantes que d'autres langues, ce qui n'est pas très égalitaire. Enfin, du point de vue de l'image de marque, l'identité visuelle de cette entreprise n'est pas respectée.

Je ne veux pas nier la complexité du travail ayant permis la création de cette

page. Elle dépasse largement les connaissances et les préoccupations du lecteur de cette police d'assurances. Le simple fait de pouvoir mettre en forme un texte de droite à gauche a constitué un grand pas en avant dans le domaine de l'informatique, sans parler des autres phénomènes mystérieux qui opèrent ici pour composer dans toutes ces langues.

Les pages typographiques multilingues, ou polyglottes, peuvent être pires que cela. Le pire cauchemar d'un auteur serait que son lecteur tombe sur une page qui ressemble à celle présentée en figure 13.

FIGURE 13 – Du tofu

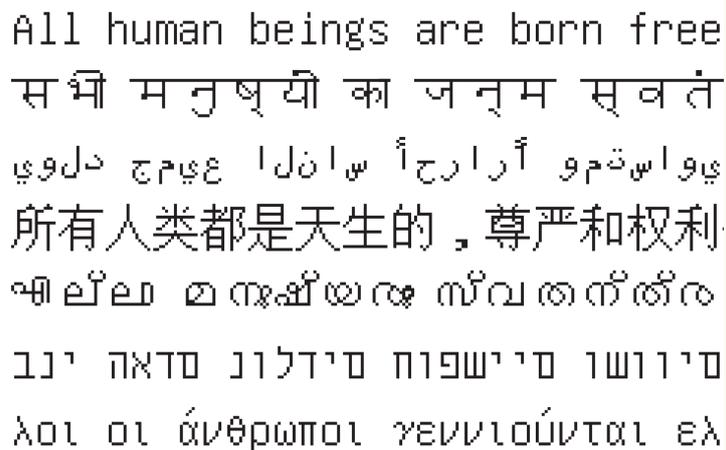


Cette page est pleine de glyphes manquants. Cette image m'a été transmise par Bob Jung, de Google, qui au sein de cette entreprise a orchestré une grande partie des débuts du projet Noto. S'il manque à un système informatique un caractère saisi par l'auteur, le lecteur voit le glyphe non défini, généralement un carré vide. Comme me l'a dit Bob, « ces carrés rappellent à beaucoup de gens, en particulier en Asie, les cubes de caillé de soja : le tofu. »

C'est de là que vient le nom des fontes Noto : pas du tofu, en anglais *No To(fu)* = Noto.

Mon voyage vers Noto a commencé en 1985, lorsque j'ai intégré l'école d'imprimerie de l'Institut de technologie de Rochester (*Rochester Institute of Technology*, ou RIT). Dans mes cours de typographie, j'ai été initié tant à la fonte chaude qu'à la fonte froide, cette dernière étant le tout dernier équipement de composition informatique de l'époque, c'est-à-dire les polices bitmap (figure 14 page suivante).

FIGURE 14 – Ma route vers Noto commence en 1985



Le professeur Archie Provan travaillait en tant que consultant pour Xerox dans le cadre de leur projet visant à rassembler des polices bitmap provenant de nombreuses fonderies différentes et capables de composer toutes les langues du monde. Il travaillait avec Ed Smura sur la norme AFII (*Association for Font Information Interchange*). Ce projet comprenait d'autres aspects liés à la typographie, notamment la classification des polices de caractères, mais le projet de polices bitmap était probablement l'aspect le plus ambitieux.

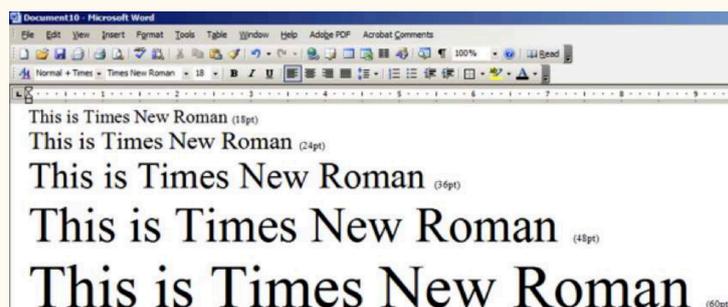
Grâce à ces polices bitmap, le système de publication Xerox Star, précurseur du Macintosh, devait pouvoir produire des documents dans n'importe quelle langue. Le rêve était que les polices soient transmises par satellite à un Xerox Star installé n'importe où dans le monde.

Lorsqu'il m'a décrit le projet, Archie m'a fait part d'une idée romantique selon laquelle, si les gens du monde entier pouvaient communiquer plus précisément et plus facilement, ils passeraient moins de temps à se battre.

Aussi banales soient-elles, ces formes bitmap monochromes allaient véhiculer la puissance de nos propres messages, de nos propres « décrets de Memphis », dans le monde entier.

Lorsque j'ai obtenu mon diplôme, les polices vectorielles que nous utilisons aujourd'hui se professionnalisaient. Les formes bitmap irrégulières de taille unique n'étaient soudain plus suffisantes, puisqu'il était désormais possible d'agrandir un caractère à l'infini pour obtenir une typographie plus expressive (figure 15).

FIGURE 15 – Premières polices de caractères vectorielles



J'ai commencé à travailler sur les polices du système TrueType pour Microsoft

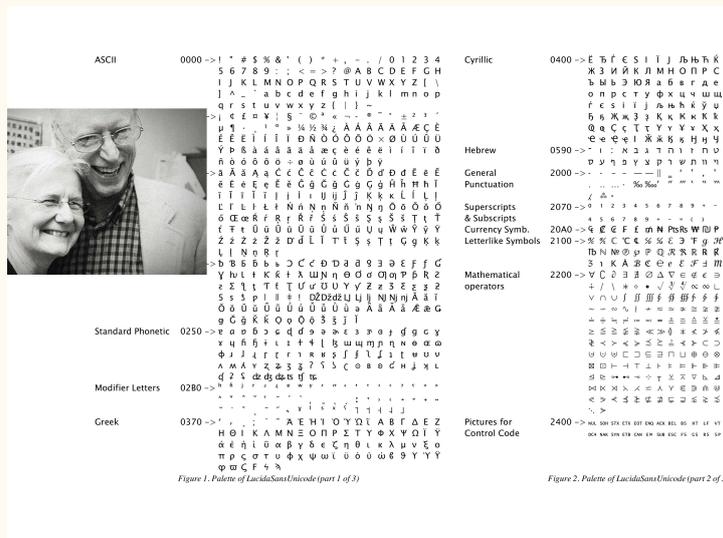
en 1990. Ces polices avaient toutes un jeu de caractères modeste d'environ 300 caractères, ce qui était déjà plus que ce que les premiers formats de polices pouvaient gérer. Les polices TrueType pouvaient gérer plus de caractères, et donc composer plus de langues, parce que l'Unicode était devenu le moyen standard de coder ou « d'organiser » toutes les lettres de tous les alphabets du monde, autorisant (à l'époque) 65 536 caractères au total. Les systèmes d'encodage précédents, y compris ISO 8859, n'autorisaient généralement que 256 caractères dans un seul fichier.

Avec Unicode, chaque caractère dans le monde reçoit un identifiant unique. Par exemple, le « G doux » du turc reçoit un nom et un numéro Unicode (Ğ, U+011E), tout comme l'omega grec (Ω, U+03A9), etc. Le registre Unicode est constamment mis à jour. En 1999, par exemple, le consortium Unicode s'est empressé de placer le symbole de l'euro (€, U+20AC) dans son répertoire afin que les fonderies de polices de caractères comme Monotype puissent mettre à jour leurs énormes bibliothèques de polices pour prendre en charge la nouvelle monnaie.

Entre 1990 et 1993, à jet continu, les fonderies ont continué à développer des polices codées en Unicode, dont la taille ne cessait de croître. Le jeu de caractères WGL (ou *Windows Glyph List*), défini par Microsoft, augmentait les attentes en matière de polices de caractères, qui devaient prendre en charge environ 90 langues, avec environ 600 caractères uniques dans un fichier de police.

Bigelow et Holmes ont créé leur police Lucida Sans Unicode pour coïncider avec la publication de la spécification 1.0 d'Unicode (figure 16). Lucida Sans Unicode a été publié par Microsoft en 1993 et a ajouté le grec, le cyrillique et l'hébreu au jeu de caractères latins. La police prenait également en charge les symboles phonétiques et mathématiques définis par l'Unicode pour la version 1.0.

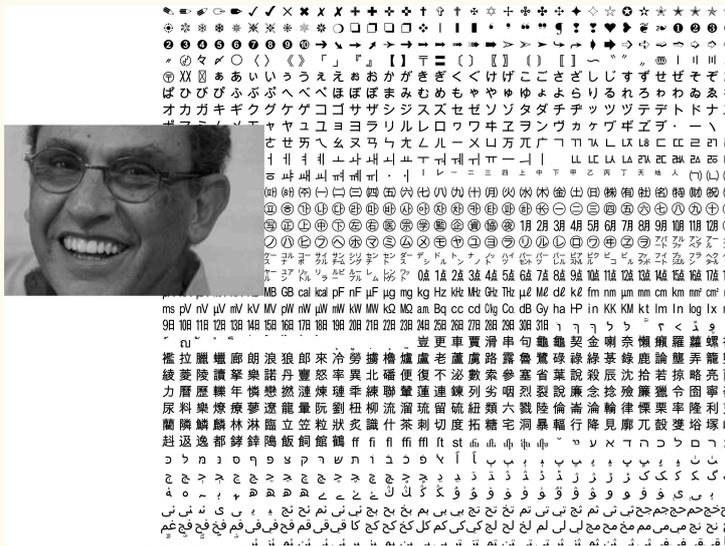
FIGURE 16 – Bigelow & Holmes Lucida Sans Unicode, 1993



En 1997, Monotype a été chargé par Microsoft d'étendre Arial à l'ensemble de l'Unicode 2.0, soit seulement 50 000 formes de lettres pour un poids de 22 Mo pour le seul fichier de police. Bien que le format de police TrueType puisse théoriquement prendre en charge tous ces caractères dans un seul fichier de police, il s'agissait d'un processus délicat à mettre en œuvre. Nous avons

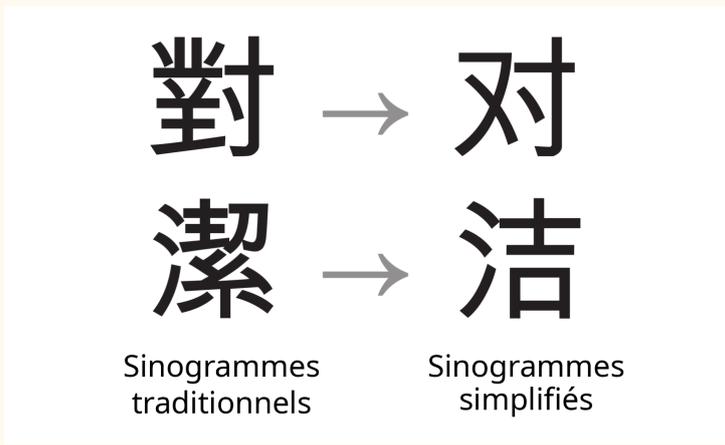
dû créer de nombreuses petites fontes (que nous appelions *fontlets*) avant de les assembler à la fin du processus. Mon collègue de longue date, Kamal Mansour, a veillé à ce que la conception soit, autant que possible pour l'époque, en harmonie avec Arial (figure 17).

FIGURE 17 – Monotype Arial pour Unicode 2.0, 1997



Nous avons alors inclus beaucoup plus d'idéogrammes chinois que ceux définis par Unicode. Les glyphes supplémentaires ont été inclus pour prendre en charge à la fois le chinois simplifié et le chinois traditionnel, de sorte que le nombre réel de formes de lettres dans la police dépasse celui de la norme Unicode (figure 18).

FIGURE 18 – Même Unicode, formes différentes



La figure 18 montre comment le même caractère Unicode peut être représenté par deux glyphes différents. Un équivalent occidental pourrait être de montrer la lettre *g* à la fois sous forme de boucle simple et de boucle double - il s'agit du même caractère Unicode représenté par deux formes de glyphes différents.

En 2006, Google m'a demandé de créer une famille de caractères pour une nouvelle plateforme de téléphonie mobile. Google voulait une interface utilisateur unique pour la marque Android. Une expérience d'interface unique commence, bien sûr, par le type de police de caractères avec lequel interagir.

En raison de l'image de marque quelque peu décalée de Google et d'Android,

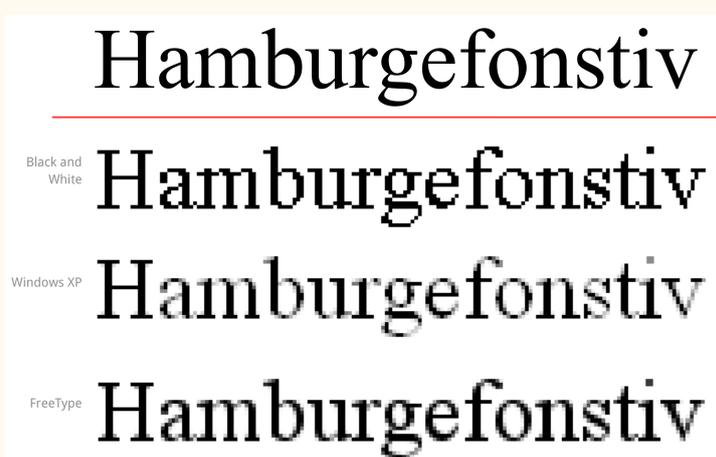
j'avais besoin de dessiner une police de caractères accessible qui le soit aussi; une fonte biaisant avec la neutralité : si elle était trop neutre, elle ne se démarquerait pas comme étant unique à la marque. Mais si elle était trop mignonne ou trop techno, la lisibilité et la fonctionnalité en pâtiraient (figure 19).

FIGURE 19 – Nouvel objectif de création de caractères pour Android



Android disposait également d'une interface graphique très spécifique. Rappelons que les écrans des téléphones portables étaient loin d'avoir la résolution des appareils d'aujourd'hui (figure 20).

FIGURE 20 – Différents écrans, différents résultats, 2006



L'exemple en figure 21 montre qu'un même dessin peut avoir un aspect très différent selon l'écran sur lequel il est visualisé. Les parties étroites des lettres peuvent sembler disparaître ou être légèrement trop lourdes, selon la manière dont le logiciel interprète et dessine les lettres. Cet exemple présente quelques dessins et expériences préliminaires qui furent testés.

FIGURE 21 – Premiers dessins pour la police Android

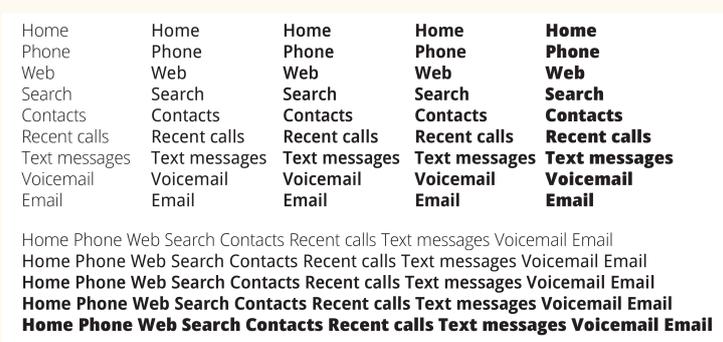


Nous avons dû créer des polices de caractères et les installer sur des appareils pour observer les effets de petits changements dans la conception et la proportion des lettres individuelles. Il s’agissait d’un processus laborieux car l’interface utilisateur était conçue en même temps que la police de caractères. Pour compliquer encore les choses, le matériel qui allait devenir les premiers téléphones Android était également en cours de conception et de fabrication.

J’ai conçu les polices en tenant compte de la façon dont elles s’afficheraient sur les différents écrans Android. J’ai travaillé avec l’équipe chargée de l’interface utilisateur pour m’assurer qu’il y avait suffisamment de contraste entre les graisses normales et les graisses grasses afin d’établir une hiérarchie dans l’interface. Je me suis assuré que les détails de chaque *design* fonctionnaient bien dans ces résolutions limitées.

Une fois que nous avons été sur la bonne voie avec le *design*, nous avons fait des tests de graisse pour voir quel contraste était nécessaire entre le texte courant et le gras (figure 22).

FIGURE 22 – Tests de graisse pour la police Android



Seules deux graisses étaient nécessaires ; les italiques seraient synthétisées par Android afin d’économiser de l’espace de stockage. D’autre part, après de nombreuses discussions, il a été décidé qu’une police avec empattements devait faire partie de l’ensemble des polices de base. Les polices avec empattements serviraient à la lecture des fils d’actualité et des textes longs. Les polices sans empattements seraient destinées aux éléments de l’interface utilisateur et aux menus.

FIGURE 23 – Famille de fontes Droid originale



Toutes les polices prenaient en charge le jeu de caractères WGL-4, mais on sentait qu’il allait être étendu si Android était un succès.

L’objectif était bien sûr de créer une famille de polices qui tienne la route sur des écrans de petite taille et qui donne à la plateforme une apparence accessible et conviviale. Nous y sommes peut-être parvenus, car un rédacteur de l’édition en ligne du magazine *Wired* a qualifié les polices de *googly*.

FIGURE 24 – Test d’affichage de Droid



Lorsque le moment est venu de décider de ce qui pouvait être affiché dans l’interface utilisateur d’Android, Google a estimé qu’il était clair qu’ils ne voulaient pas que cela ressemble à l’exemple de gauche présenté en figure 25, qui est malheureusement assez similaire à la déclaration de prestations d’assurance montrée plus haut (fig. 4). L’entreprise souhaitait plutôt une présentation harmonisée de la « marque Android » dans toutes les langues qu’elle prenait en charge. Tous les scripts devraient avoir un aspect contemporain, accessible et ergonomique, plus proche de l’exemple à droite. (Toutes nos excuses pour la faute de frappe dans le texte arabe).

FIGURE 25 – Scripts non harmonisés à gauche ; harmonisés à droite



Concevoir des formes cohérentes pour des fontes qui ne possèdent pas de sources historiques communes est quelque peu délicat et dans certains cas il faut adopter un compromis pour qu’elles affichent une grasse similaire. Par exemple, les pleins des caractères arabes sont sur les horizontales alors qu’ils se placent sur les fûts verticaux des alphabets latins. Cette particularité impose une énorme différence de réglage des graisses pour les équilibrer. L’harmonisation visuelle peut être obtenue en exportant des particularités graphiques des formes latines telles que la forme, la grasse de l’extrémité des empattements et des terminaisons ou l’équilibre entre les pleins et les déliés.

FIGURE 26 – Harmonisations entre l'arabe et le latin

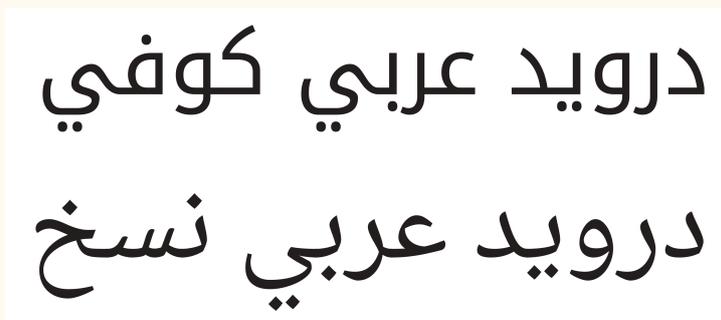


Les polices arabes ont été conçues par Pascal Zoghbi avec, de ma part, un peu de direction artistique. La graphie qui correspond le mieux aux caractères latins à empattements est appelée naskh. Dans la plupart des caractères naskh, les contreformes sont minuscules mais, dans ce cas, nous avons exagéré leur taille pour imiter celles des caractères latins et faciliter la lisibilité à l'écran.

Dans l'exemple figure 27, la ligne supérieure du texte arabe est composée dans un style appelé kufi (en français, il est parfois appelé coufi, ou coufique) qui se rapproche davantage d'une fonte latine sans empattements. Toutefois, les lecteurs arabes préfèrent généralement le style naskh (que l'on voit dans la ligne inférieure) pour une lecture prolongée. J'ai l'impression que cela ressemble à la résistance qu'avaient les lecteurs occidentaux à lire des livres écrits avec des caractères sans empattements. Ce n'est que dans les années 1950 et 1960 que les gens ont commencé à accepter cette nouvelle approche typographique.

En travaillant sur ce projet, j'ai appris que les lecteurs arabes étaient très habitués à devoir zoomer sur le texte dès qu'une page était chargée. L'un des objectifs de l'exagération des proportions était d'éviter d'avoir à zoomer pour lire les tailles arabes par défaut. D'après ce que j'ai compris de Google, nous y sommes parvenus avec Droid et Noto.

FIGURE 27 – Style kufi (en haut), style naskh (en bas)



Au-delà de l'arabe, il était urgent d'équiper Android de caractères thaïlandais. Il s'agit d'une fonte à empattements et l'on peut noter que certains détails font penser à la fonte latine du même type (l'exemple figure 28 page suivante et

FIGURE 31 – Droid arménien

րդ էջերում: Պ
 առշումն ընդու
 պետությունն է
 report Հռչակս

De même, le géorgien s’inspire des formes rondes du latin. Sa texture est totalement différente de celle de l’arménien, mais il appartient clairement à la famille des polices de caractères Droid.

FIGURE 32 – Droid géorgien

1948 ვინაიდან ადამია
 კრისათვის Australian
 ანასწორი და განუყოფ
 nyon აღიარება წარმო
 მართლიანობის და va
 ვინაიდან ადამიანის უ
 გულებელყოფამ და აბ

Certains se demanderont peut-être pourquoi je parle d’Android à ce stade de mon voyage vers Noto.

FIGURE 33 – Les travaux se succèdent

	Droid Sans	Droid
	chrome	Open Sans
	Noto Sans	Noto

En 2009, on m’a demandé d’adapter Droid Sans pour en faire un caractère de marque pour Chrome. Ce dessin est devenu Open Sans, une version légèrement plus large du Droid, plus adaptée au texte normal dans les documents. Parallèlement, les équipes Chrome et Android de Google discutaient de l’idée d’unir leurs efforts pour développer une super police qui couvrirait l’ensemble de l’Unicode. Le problème de l’affichage de « tofu » dans les recherches sur internet devenait de plus en plus problématique à mesure que l’utilisation d’internet dans le monde augmentait de façon spectaculaire.

Avec l’arrivée conjointe de Chrome et d’Android, Open Sans est devenu la base de Noto Sans et Droid Serif est devenu Noto Serif.

En 2011, il a été décidé d’élargir la famille latine de Noto pour qu’elle comprenne une gamme typographique complète de familles, de graisses et de corps. Les versions condensées et étroites des caractères, avec et sans empattements, ont été ajoutées, ainsi que des graisses supplémentaires allant de l’ultra-léger à l’extra-noir. La fonte avec empattements a été enrichie au point d’offrir une gamme de gris typographiques allant du très dense au très clair. Cela représente environ 72 styles de police par famille — la palette typographique polyglotte n’est plus limitée aux seuls styles réguliers et gras !

FIGURE 34 – Noto, vers 2011



Noto Sans **Noto Sans**
Noto Sans Noto Sans
 Noto Serif **Noto Serif**
Noto Serif Noto Serif
 Noto Sans Monospaced

À cette époque, l’Unicode compte environ 3400 caractères pour prendre en charge l’écriture latine, cyrillique, grecque et phonétique. Cet ensemble de caractères permet de prendre en charge plus de 500 langues. Avec l’ajout de ces caractères supplémentaires, toutes les familles, graisses et corps des fontes latines supplémentaires signifiaient que l’on dessinait environ 230 000 caractères.

Il était désormais évident que Noto ne serait jamais livré sous la forme d’une seule police binaire avec l’ensemble de l’Unicode.

Rappelez-vous que l’ancienne police Unicode Arial pesait à elle seule 22 Mo et ne contenait que 55 000 caractères. Au lieu de cela, les polices Noto sont construites comme des modules individuels couvrant une ou quelques gammes de scripts Unicode, ou systèmes d’écriture.

Outre le problème de la taille des fichiers, certains systèmes d’écriture deviennent assez complexes dans leur forme, bien au-delà du thaï que nous avons dessiné. Verticalement, le thaï est tout juste capable de se glisser dans la métrique verticale d’une police de caractères latine. En figure 35, vous

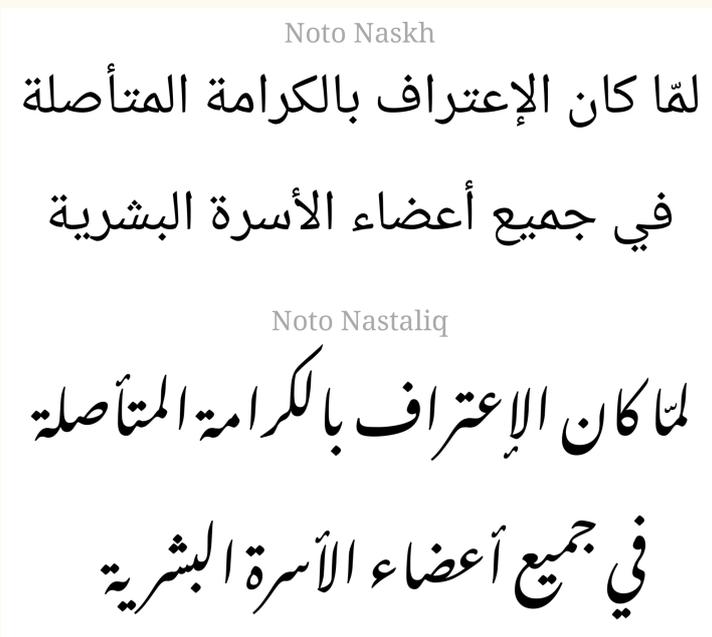
pouvez voir que le javanais (à gauche) et le khmer (à droite) sont encore plus complexes verticalement, et ne peuvent tout simplement pas raisonnablement entrer dans les dimensions de l'écriture latine.

FIGURE 35 – Au-delà de la métrique latine : javanais (à gauche), khmer (à droite)



Un autre exemple est le style nastaliq de l'arabe utilisé pour l'ourdou, le pachtoune et le persan. Alors que le naskh, couramment utilisé, suit une ligne de base horizontale, la ligne de base du nastaliq est inclinée vers la gauche. Plus le mot est long, plus la ligne descend.

FIGURE 36 – Noto naskh (ligne de base plate) et Noto nastaliq (ligne de base inclinée)



L'une des façons dont nous avons envisagé cet énorme projet a été de diviser les systèmes d'écriture en catégories. De cette manière, nous pouvions classer les écritures apparentées en fonction de leur région, de leur complexité ou de leur utilisation relative dans le monde moderne.

Les cases oranges de la figure 37 page suivante indiquent des écritures qui peuvent être mortes depuis des centaines d'années ou plus, tandis que les vertes sont des écritures utilisées en Inde, et ainsi de suite. Cela a permis de

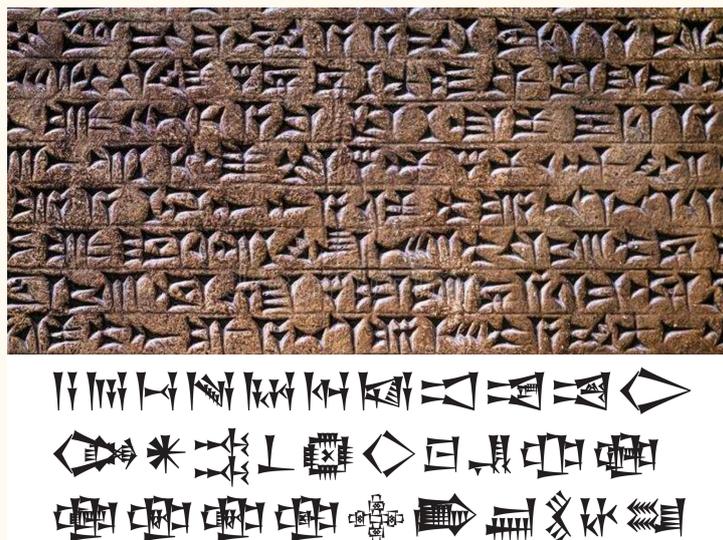
hiérarchiser et d’organiser l’expertise nécessaire à la réalisation de chaque élément du projet.

FIGURE 37 – Organisation des écritures pour Noto



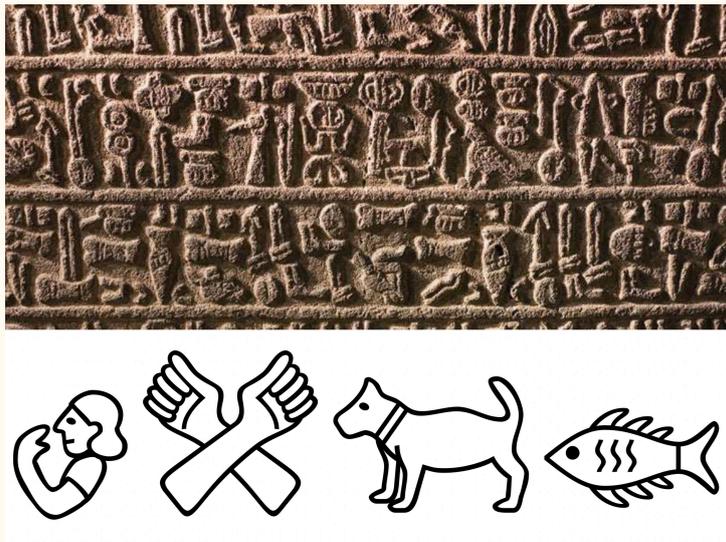
Les caractères cunéiformes sont un bon exemple des « écritures mortes » que nous venons de mentionner ; ils peuvent avoir jusqu’à 5000 ans. Bien qu’ils ne soient pas utilisés dans la pratique, ils sont certainement utiles pour les chercheurs et les linguistes de les avoir codés dans un fichier de polices de caractères.

FIGURE 38 – Cunéiforme, original et Noto



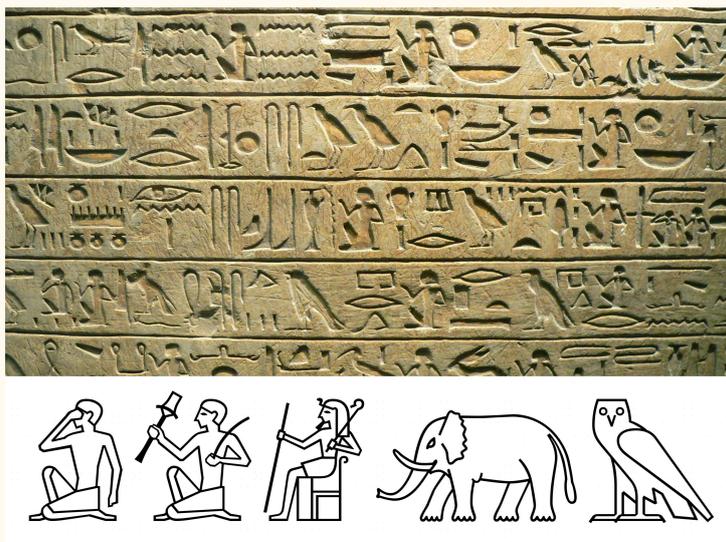
Les hiéroglyphes anatoliens datent d’au moins 4000 ans et sont donc antérieurs aux hiéroglyphes égyptiens. Ils sont donc représentés comme légèrement plus bruts et lissés.

FIGURE 39 – Hiéroglyphes anatoliens, originaux et Noto



Les hiéroglyphes égyptiens sont beaucoup plus nets et raffinés que leurs ancêtres anatoliens.

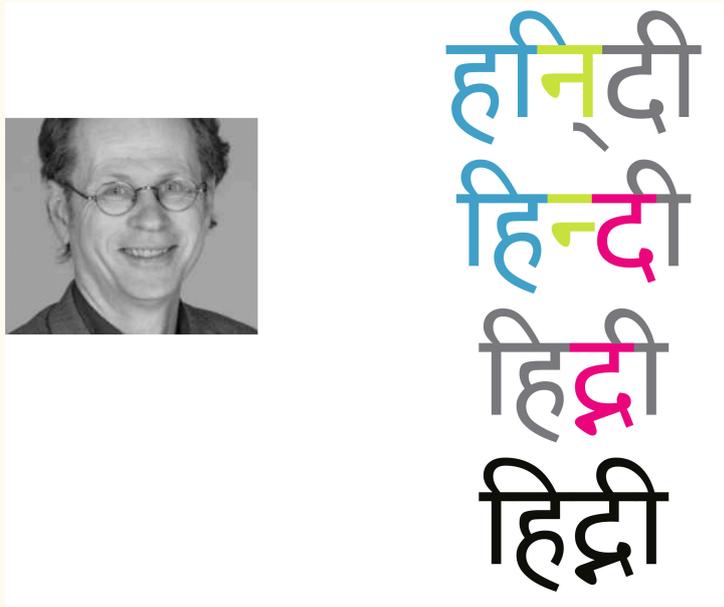
FIGURE 40 – Hiéroglyphes égyptiens, originaux et Noto



De nombreuses écritures sont classées comme « complexes » et nécessitent une grande quantité de programmation pour assembler les mots de la bonne manière. L'arabe, qui se lit de droite à gauche et comporte de nombreuses formes de la même lettre, entre dans cette catégorie, tout comme les écritures utilisées en Inde. Jelle Bosma est le responsable créatif de Monotype pour la création des écritures indiennes et travaille sur les mises à jour de la version 13 d'Unicode.

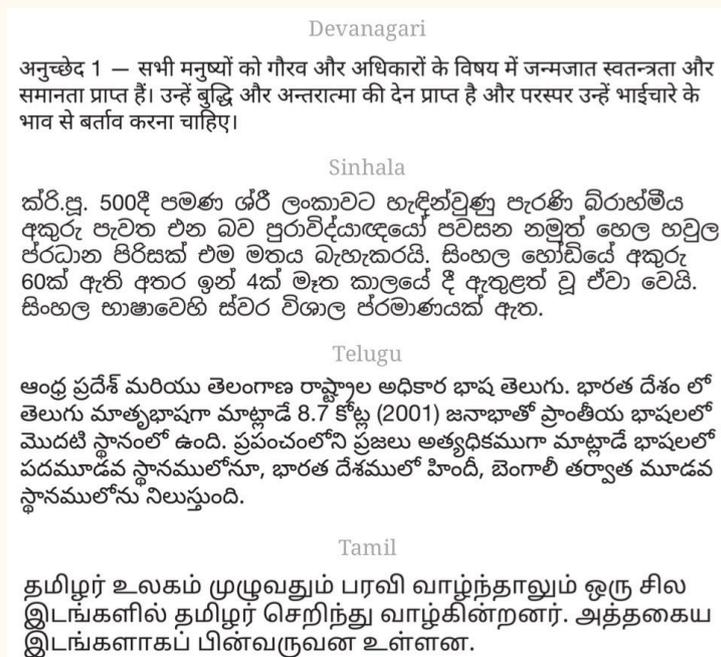
Dans l'illustration figure 41 page suivante, le mot de droite est le mot « Hindi » en écriture devanagari. La ligne supérieure montre l'aspect du mot avec des caractères Unicode simples assemblés pour épeler le mot. Chacune des lignes successives montre comment l'écriture est « remodelée » grâce à des tables typographiques avancées qui réorganisent les lettres au fur et à mesure qu'elles sont tapées. Sur la deuxième ligne, les caractères verts changent de place. Sur la troisième ligne, les caractères verts et roses forment une ligature.

FIGURE 41 – À gauche, le graphiste Jelle Bosma; à droite, le mot « Hindi » est formé au fur et à mesure de la frappe



Les systèmes d'écriture indiens semblent très différents les uns des autres, mais nous les avons conçus pour qu'ils s'harmonisent autant que possible. Le devanagari, assez rectangulaire, contraste beaucoup avec le cinghalais, qui est plus fluide, mais leur couleur et leur proportion sont préservées pour les maintenir en phase. Il en va de même pour le télougou et le tamoul, dont la texture générale est très différente, mais leurs proportions et leur couleur les relient.

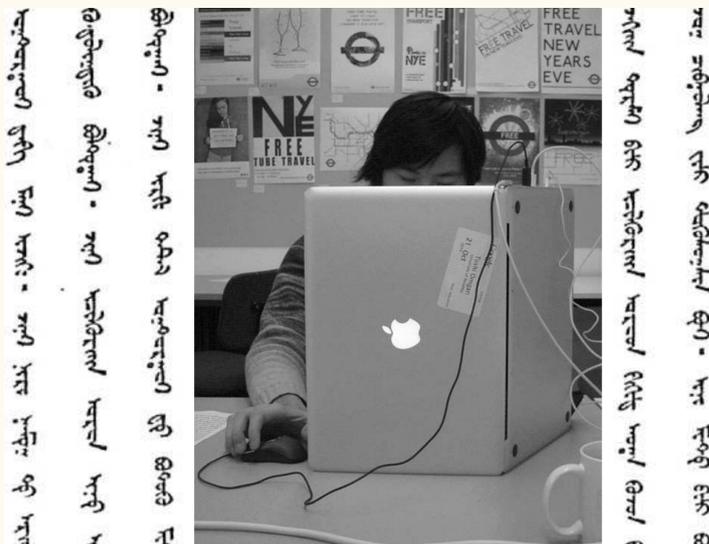
FIGURE 42 – Des écritures indiennes composées avec Noto



Tout comme j'ai mentionné qu'il y avait des problèmes techniques à résoudre avec l'Unicode Arial, il en va de même avec Noto. Mon collègue Toshi Omagari est représenté ci-dessous avec son ordinateur portable tourné de 90 degrés, ceci pour dessiner le mongol, une écriture connectée qui se lit de haut en bas, de gauche à droite.

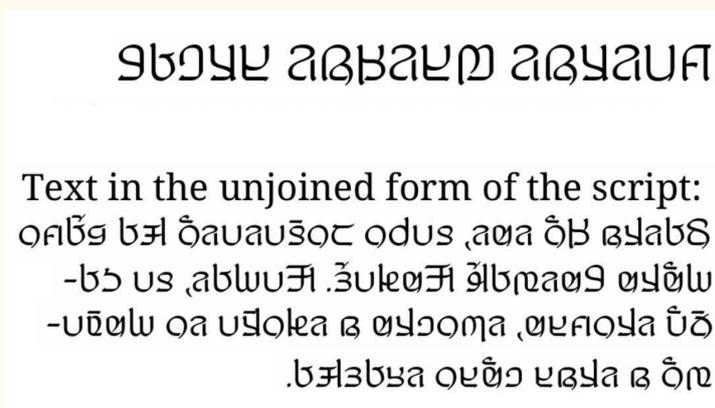
Les développeurs de l’outil GlyphsApp ont été d’un grand soutien en mettant à jour leur logiciel afin de rendre possible ces scripts complexes dans Noto. Il n’a pas fallu longtemps pour qu’ils fournissent une correction permettant à Toshi de voir son travail de la manière dont il serait utilisé.

FIGURE 43 – Toshi Omagari travaillant sur le mongol



J’ai parlé plus haut de « écritures mortes ». À l’opposé, il y a l’Adlam, une écriture développée à la fin des années 1980 par les frères Ibrahima et Abdoulaye Barry. Ce système d’écriture transcrit la langue peul parlée en Guinée, au Nigeria et au Liberia. Avant l’Adlam, le peul était écrit en arabe ou en latin.

FIGURE 44 – Adlam, développé dans les années 1980



En 2018, les polices Noto couvraient près de 64 000 caractères et de nombreuses mises à jour ont été effectuées au cours des deux dernières années. L’Unicode 13.0 ajoute 4 nouvelles écritures et 5000 nouveaux caractères à ce décompte. « Que reste-t-il encore ? » pourriez-vous vous demander.

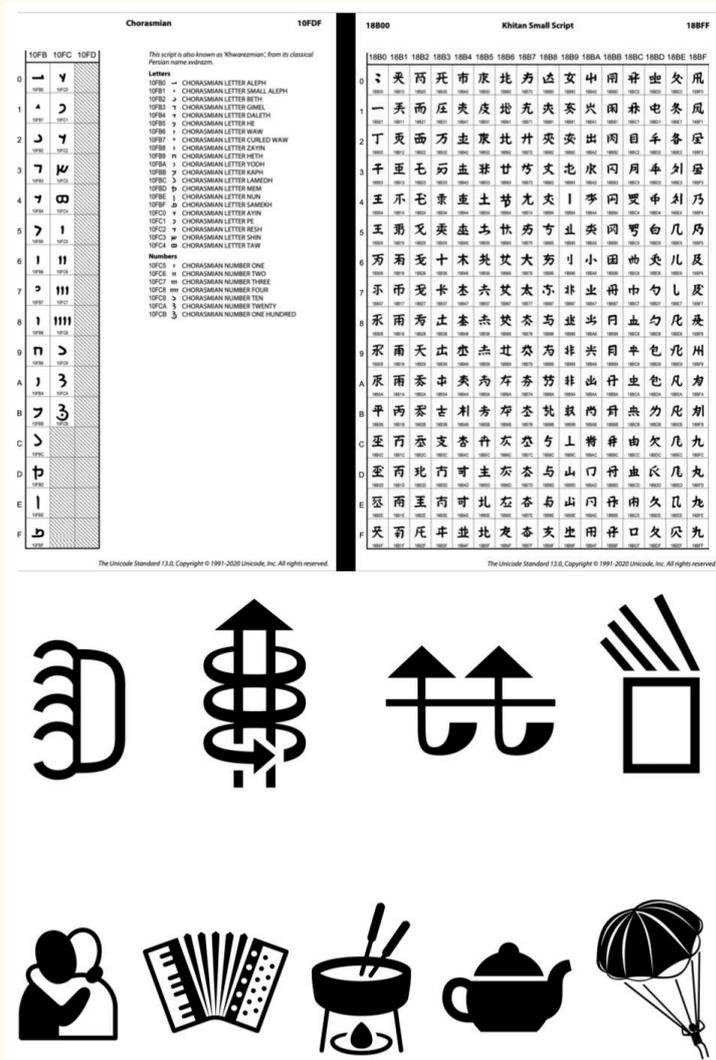
Le khitan, une langue autrefois parlée en Mandchourie, a été ajouté, de même que le chorasmien, une langue de l’ancienne Perse.

SignWriting⁵¹, un système de notation utilisé pour enseigner la langue des signes, a été ajouté. Il nécessite des milliers d’icônes pour représenter les gestes de la main et les expressions faciales utilisés par les interprètes en langue des signes.

51. <https://fr.wikipedia.org/wiki/SignWriting>

La police de symboles Noto s’enrichit de nombreux nouveaux caractères, dont les symboles tant attendus de l’accordéon et de la fondue ⁵². La liste ne cesse de s’allonger.

FIGURE 45 – Ajoutés dans Unicode 13.0 : écritures khitan et chorasmienne, ou khwarezmienne, (en haut), SignWriting (au milieu) et différents symboles récents (en bas)



Noto continuera d’être améliorée pour refléter les demandes des différentes communautés humaines qui utilisent ces fontes.

Outre la trentaine de personnes de Monotype qui ont travaillé sur le projet Noto, nous avons collaboré avec plus de 30 linguistes, consultants et concepteurs extérieurs dans le monde entier, notamment Fiona Ross, Tiro Typeworks et Kigali Design.

J’aimerais également remercier d’autres personnes : Abdoulaye & Ibrahima Barry; Jo De Baerdemaeker; Cadson Demak Ltd; Diane Collier; Fontef Type Foundry; Kalapi Gajjar-Bordawekar; Yanone Gerner; Gajjar & Vilhjamsson Private Limited; Kimya Gandhi; Patrick Giasson; John Hudson; Indian Type Foundry; Yanek Iontef; Letterjuice Ltd.; Ben Mitchell; James Montalbano; Elena Papassissa; Rainer Erich Scheichelbauer; Zachary Scheuren; Georg Seifert; Vaibhav Singh; Anuthin Wongsunkakon; Pascal Zoghbi.

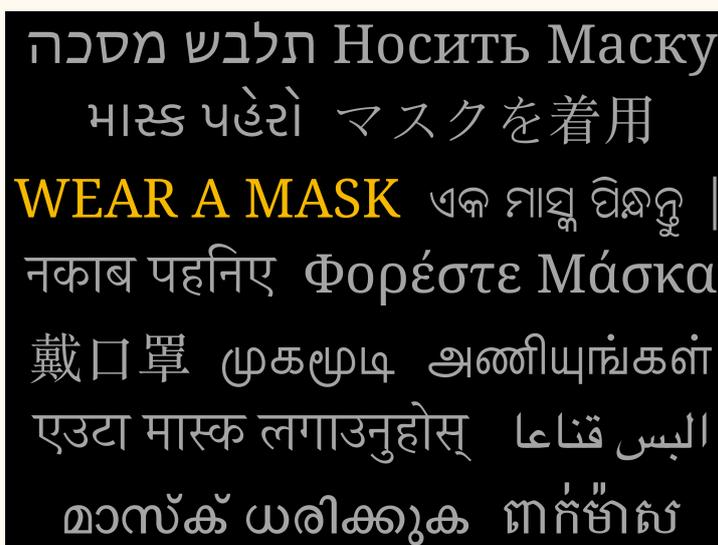
La communauté des utilisateurs et testeurs qui ont fourni un retour d’in-

52. Quelle fondue, l’auteur ne le dit pas. La rédaction le regrette

formation est, bien entendu, beaucoup plus importante. C’est un honneur de travailler pour eux, de faire de cette énorme entreprise un outil à leur disposition.

En fin de compte, Noto ne sera peut-être pas utilisé pour retraduire la Bible ou des décrets impériaux. Il pourrait s’agir de messages très simples que nous pouvons transmettre avec cet énorme outil dont nous disposons. Et peut-être que Noto est un pas de plus vers la réalisation concrète de cette idée romantique que j’ai entendue lorsque j’étais étudiant, à savoir que les gens du monde entier pourraient passer plus de temps à communiquer plutôt qu’à se battre.

FIGURE 46 – Traductions non retouchées, via Google Translate



Steven Matteson

Traduction Patrick Bideault

Le traducteur remercie pour leur expertise Alain Joly et Claude Leloup, typographes.

