

Démystification des QR codes

Que ce soit sur les affiches publicitaires, sur les bouteilles d'eau minérale, sur des cartes de visite, dans les arrêts de bus ou même sur les pages de ce magazine où ils renvoient vers des contenus complémentaires, les QR codes ont envahi notre quotidien.

En première approche, les QR codes ressemblent à des carrés composés de pixels blancs et noirs, qu'on appelle « modules », dont la quantité et la répartition représentent le message contenu dans le code. C'est en quelques sortes une « évolution » des classiques codes-barres (en une dimension) et des codes Data Matrix (deux dimensions) notamment présents sur les emballages des médicaments.

Un peu d'histoire

Initialement, les QR codes (pour Quick Response) ont été créés en 1994 par Denso-Wave pour le compte de la société Toyota. Cette dernière avait besoin d'un support simple, petit et puissant pour suivre les pièces dans ses usines automobiles.



QR code contenant le message
« Le portail des développeurs »

Le standard AIM (Automatic Identification and Mobility) a été publié en 1997. Deux ans plus tard, l'entreprise Denso a rendu libre l'utilisation du QR code et a publié le standard japonais JIS XO 510. Dans les années qui suivent, et jusqu'en 2009, plusieurs normes ISO (ISO/CEI 18004) viennent encadrer les spécifications du QR code et ses évolutions.

Utilisation possibles

L'utilisation la plus courante des QR codes est sans doute d'envoyer l'utilisateur vers un site Web. Certains codes renvoient plus spécifiquement vers une page précise du site, correspondant par exemple à la description d'un produit commercial. On retrouve de plus en plus cette pratique dans les catalogues des supermarchés. Ainsi, lorsqu'on scanne le code associé à un produit, on est redirigé vers la liste des caractéristiques, avec des photos additionnelles, des avis d'utilisateurs, des codes promotionnels, etc.

Les offices du tourisme, les sites nationaux mais aussi les musées proposent également de scanner des QR codes. Cela permet d'effectuer une visite personnalisée, enrichie de nombreux contenus (Web, audio, vidéo, etc.), plus agréable qu'avec les anciens guides audio. En arrivant devant un tableau ou une sculpture, on obtient des informations sur l'œuvre, sur son auteur, des anecdotes, etc. Les possibilités sont nombreuses.

« Une visite enrichie de contenus... »

De façon un peu détournée mais très appréciée, de nombreux arrêts de bus proposent désormais de scanner des QR codes pour connaître les horaires de la ligne et l'heure de passage du prochain bus. Cette dernière est réactualisée en permanence en fonction de la position du véhicule, du trafic, des statistiques de circulation, etc. Pour cela, le QR code renvoie simplement vers une URL dont un des paramètres correspond à l'identifiant de la station. Cette URL est fixe mais le contenu de la page, lui, est dynamique.

Les QR codes contiennent simplement un message. Ce qui différencie les différents types de code, c'est le contenu du message, et plus précisément sa structure. Ainsi, certains QR codes contiennent des cartes de visite au format vCard ou meCard (cf. signature de cet article), des paramètres de connexion automatique à des réseaux WIFI, des numéros de téléphone qui provoquent un appel lors du scan ou l'envoi d'un SMS, des dates d'événement (salon, concert, match de foot...) automatiquement ajoutés dans l'agenda, etc.

Pour réaliser l'une ou l'autre de ces actions, le message doit contenir des marqueurs spécifiques. Par exemple, pour lancer un appel, il faut utiliser le marqueur « tel ». Pour un email, on utilisera « mailto » comme on le ferait en HTML :

```
String email = "jean.dupont@hotmail.com";  
String sujet = "Vacances";  
String texte = "Bonjour%20bla%20bla";  
String message = "mailto:" + email  
+ "?subject=" + sujet + "&body=" + texte;
```

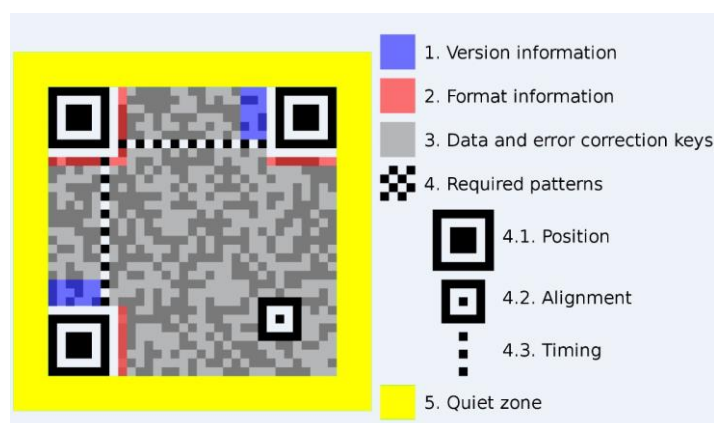
Il est possible d'utiliser des marqueurs propriétaires. La seule limitation est que le scanner sache les reconnaître. Ainsi, certaines sociétés de transport (train, avion, etc.) impriment des QR codes sur les tickets mais on ne peut les décoder que si on dispose du bon logiciel ; le contrôleur en est évidemment équipé.

Structure des QR codes

Les QR codes sont constitués de modules (cf. image). Certains sont fixes : yeux de positionnement (modules entourés d'un carré et placés dans les coins), yeux d'alignement (petits yeux) utilisés pour corriger la perspective induite par le scanner, timing (barres en pointillés joignant les yeux de position), etc. Ces modules fixes permettent au scanner de déterminer l'orientation du QR code et d'en établir le quadrillage.

Les autres zones contiennent le message codé, de la réplique de données ou encore des informations techniques (elles aussi redondées) : numéro de version, format utilisé, encodage, masque, etc.

Enfin, bien que ce ne soit pas absolument indispensable en pratique, les QR codes sont entourés d'une zone vide (blanche) qu'on nomme « quiet zone ». Cet espace sert de délimitation avec le reste du document (sur lequel le code est imprimé) pour faciliter la lecture. On notera simplement que l'efficacité de certaines applications de lecture diminue fortement lorsque la quiet zone est réduite ou absente. La norme prévoit qu'elle soit large de quatre modules.



Structure d'un QR code

La taille du QR code, exprimée en nombre de modules, dépend de la nature (ISO, UTF-8, chiffres, etc.) et de la quantité d'informations à coder ; plus le message est long et plus il faudra de modules pour le représenter. Ainsi il faudra beaucoup plus de modules pour représenter une vCard (carte de visite

avec de nombreux champs) qu'une simple URL. Les spécifications prévoient quarante versions (tailles) allant de 21 modules de côté à 177.

Dans sa taille la plus petite, un QR code ne peut contenir que 25 caractères alphanumériques. Dans sa plus grande taille, en fonction de l'encodage utilisé, un QR code peut contenir 7089 chiffres, ou 4296 caractères alphanumériques, ou 2953 octets ou encore 1817 kanjis. Cela nécessite alors 46 yeux d'alignement et 62 modules de timing. A titre de comparaison, un code barre ne permet de stocker qu'une vingtaine de caractères.

Le QR code contenant ma carte de visite (cf. signature de cet article) n'utilise que six yeux d'alignement. Les yeux de position, quant à eux, sont toujours au nombre de trois et disposés dans les coins.

Tout comme le Compact Disk (CD), pour lequel Philips a prévu qu'une partie des données du support puisse être détruite (le CD peut supporter des rayures d'un millimètre d'épaisseur) car elles sont dupliquées ailleurs sur le disque, le QR code intègre une certaine redondance de l'information. Le QR code reste ainsi lisible même s'il est déchiré ou recouvert d'une tache d'encre. La norme prévoit plusieurs niveaux de « qualité » allant de L (Low-7% de redondance) à H (30%) en passant par M (Medium-15%) et Q (25%). Lorsqu'on emploie le niveau H (High), cela signifie que 30% de la surface utile du QR code correspond à de la redondance d'informations.

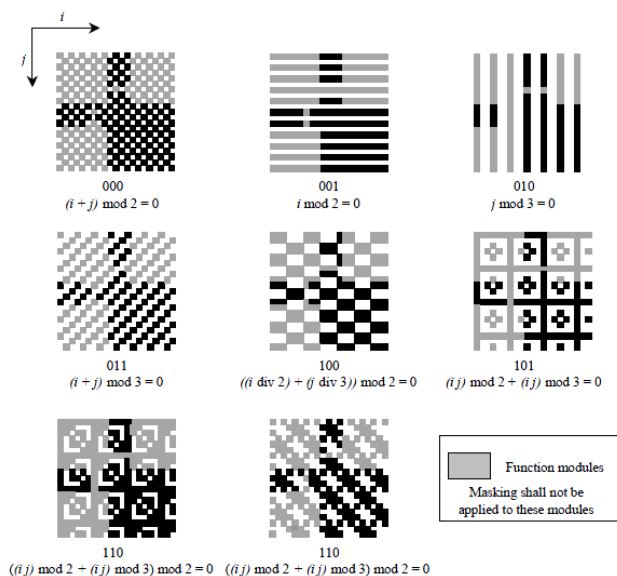
Pour un message donné, cela veut aussi dire que plus le niveau de qualité est élevé et plus le nombre de modules nécessaires est grand. Pour une taille d'impression précise (par exemple 3 cm de côté), un niveau de qualité fort entraîne donc l'utilisation de modules plus petits. Or plus les modules sont petits et plus la caméra (lecteur du smartphone) aura du mal à les scanner. De manière générale, l'étalement des paliers des niveaux de qualité fait que le niveau Q (Quality) est souvent un bon compromis.

« On préfère les modules uniformément répartis... »

L'algorithme de génération des modules peut conduire à des images laides ou illisibles. Le principal mal est d'avoir un pâté de modules (noirs) agglomérés dans lequel on a du mal à compter le nombre exact de modules verticaux et/ou horizontaux. Y en a-t-il quatre ou cinq ? Une autre anomalie courante est de générer des motifs

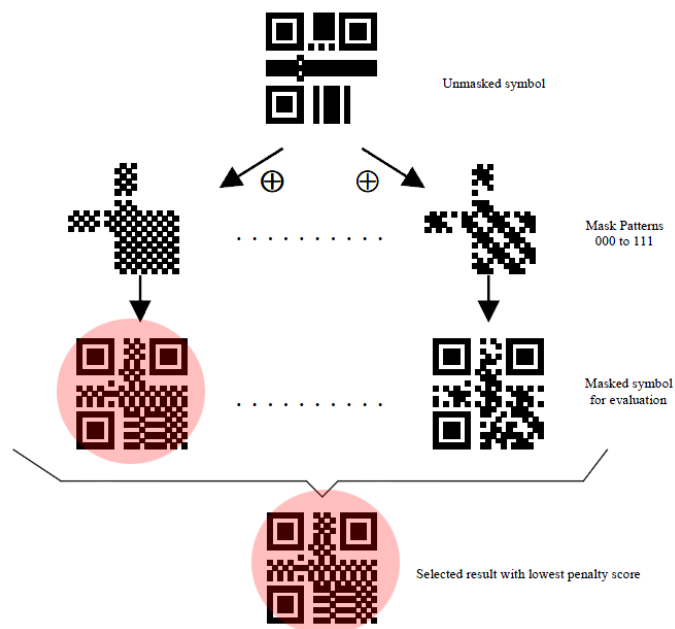
géométriques interdits, comme des yeux. Un œil supplémentaire empêcherait à coup sûr de déterminer l'alignement du code. Enfin, et pas seulement pour des raisons esthétiques, il est préférable que les modules soient uniformément répartis sur la surface.

Pour lutter contre ces anomalies, on superpose un masque (calque), possédant des modules bien placés, par-dessus le QR code et on conserve les modules résultant d'une opération booléenne XOR entre le masque et le code. La norme prédéfinit huit calques (cf. image) répondant à des fonctions mathématiques simples, à base de modulus.



Masques prédéfinis

Lors de la génération, le programme doit tous les essayer et conserver celui qui donne le plus beau résultat. Plus exactement, on garde celui qui est le moins laid à l'aide d'algorithmes de pénalités. Le moins laid est celui qui obtient la pénalité la plus faible. L'identifiant du masque choisi (« 000 » dans l'exemple) est bien entendu inscrit dans le QR code.



Utilisation des masques

La bibliothèque la plus populaire pour créer des QR codes (mais aussi des codes barres, des Data Matrix, des EAN, des ITF, etc.) est sans doute Zxing. Elle permet de générer la matrice des modules en quelques lignes :

```
final String message
    = "http://www.programmez.com";

// Creation de la matrice
final QRCode qrCode = new QRCode();
Encoder.encode(message, Q, qrCode);
final ByteMatrix matrix = qrCode.getMatrix();
```

A partir de cette matrice booléenne, on peut créer l'image finale du QR code à l'aide d'une API graphique comme AWT ou Java 2D :

```
// Modules
final Area area = new Area();
final Area module
    = new Area(new Rectangle.Float(0, 0, 1, 1));
final AffineTransform transform
    = new AffineTransform();
final int width = matrix.getWidth();

for (int i = 0; i < width; i++) {
    for (int j = 0; j < width; j++) {
        if (matrix.get(j, i) == 1) {
            area.add(module);
        }
        transform.setToTranslation(1, 0);
        module.transform(transform);
    }
    transform.setToTranslation(-width, 1);
    module.transform(transform);
}

// Taille
final int size = 400;
final double adjustment
    = width / (double) (width + 8);
final double ratio = size * adjustment / width;
```

```
// Quietzone
transform.setToTranslation(4, 4);
area.transform(transform);
transform.setToScale(ratio, ratio);
area.transform(transform);

// Traitement de l'image
final BufferedImage image
    = new BufferedImage(size, size, TYPE_INT_RGB);
final Graphics2D graphic
    = (Graphics2D) image.getGraphics();
// Modules bleus
graphic.setPaint(new Color(0x0000FF));
// Fond blanc
graphic.setBackground(new Color(0xFFFFFFFF));
graphic.clearRect(0, 0, size, size);
graphic.fill(area);
```



QR code créé par le programme

Des codes design

De plus en plus, on voit apparaître des QR codes personnalisés, utilisant des modules arrondis, des yeux géométriques (formes diverses), des jeux de couleurs, des dégradés, etc. Les exemples les plus complexes intègrent même des logos ou des graphiques.

« La France est bien représentée... »

Sur ce point, la France est relativement bien représentée puisqu'Unitag, une société leader dans la génération et la personnalisation des QR codes, est implanté à Toulouse. Le code Java proposé plus haut a d'ailleurs été gracieusement offert par leur équipe de développement.

Il est possible de changer la couleur des modules car c'est le contraste avec le fond du QR code qui compte. Ainsi, sur un fond blanc, la plupart des teintes peuvent convenir à condition de ne pas être trop claires.

On peut également insérer un petit logo dans le QR code sous réserve d'utiliser un niveau de qualité

élevé (Q ou H). Les modules corrompus par cet ajout sont compensés par la redondance des données. Il convient néanmoins de ne pas modifier les yeux de position et les zones techniques (version, format, masque, etc.) et de tester la lisibilité. Si besoin on pourra déplacer le logo dans le code pour qu'il soit plus lisible et plus esthétique.



QR code design en forme de lapin

Enfin, il est possible de créer des QR codes utilisant des modules blancs sur fond noir (ie. des pixels clairs sur fond sombre). Ils sont alors appelés « codes inversés ». Toutefois, tous les lecteurs ne savent pas forcément les interpréter.

Détection et correction d'erreur

Contrairement au protocole TCP/IP (Transmission Control Protocol/Internet Protocol), qui permet de détecter des erreurs de transmission mais qui nécessite l'envoi d'une nouvelle trame pour avoir l'information correcte, le QR code intègre un mécanisme de correction. Lors du scan du QR code, on ne peut évidemment pas demander une réimpression de ce dernier s'il contient une erreur, d'où l'obligation d'avoir un système de corrections.

« Simple mais d'une efficacité diabolique... »

L'algorithme utilisé est le « Reed Solomon » (du nom de ses inventeurs Irving S. Reed et Gustave Solomon) dont le Compact Disk emploie une variante. Son principe repose sur la construction de polynômes à partir de l'information source, que l'on va enrichir. C'est cet ensemble qui est alors envoyé à la place des caractères originaux. Cet ajout surcharge l'image finale du QR code mais en garantie le bon décodage.

Supposons qu'il faille protéger le triplet (a, b, c). Pour cela on va ajouter de l'information sous la forme du couple (S₁, S₂) à l'aide de fonctions mathématiques bien choisies :

$$S_1 = a + b + c$$

$$S_2 = a + 2 \times b + 3 \times c$$

Pour le triplet (12, 7, 21), on va ainsi ajouter le couple (40, 89) :

$$S_1 = 12 + 7 + 21 = 40$$

$$S_2 = 12 + 2 \times 7 + 3 \times 21 = 89$$

Imaginons que, pour des raisons diverses, la valeur lue de « b » soit corrompue ; on lit « 10 » au lieu de lire « 7 ». Le couple (V₁, V₂) résultant devrait donc être (43, 95). La différence Δ entre S₁ et V₁ quantifie l'erreur détectée :

$$\Delta = V_1 - S_1 = 3$$

Reste à savoir sur quel élément E du triplet porte cette erreur. Ici, l'algorithme suppose qu'un seul élément soit corrompu, ce qui est le plus probable.

$$E = (V_2 - S_2) \div \Delta = (95 - 89) \div 3 = 2$$

C'est donc le second élément qui possède un décalage (positif) de « 3 ». Il ne reste plus qu'à corriger les erreurs lues :

$$a = 12$$

$$b = 10 - 3 = 7$$

$$c = 21$$

En pratique, la simplicité apparente du Reed Solomon est d'une efficacité diabolique, ce qui lui a valu une adoption forte par l'industrie. Cet algorithme ne peut néanmoins pas corriger les erreurs multiples (sur plusieurs éléments, se compensant en partie) ou les divergences trop importantes. Les statistiques d'utilisation indiquent que ces cas restent relativement rares.

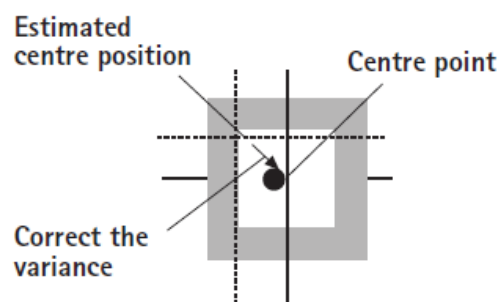
Les bonnes pratiques

La plus grosse difficulté que rencontrent les scanners est liée aux déformations de l'image. Personne ne tient son smartphone à la verticale exacte du QR code. La capture de l'image se fait toujours de manière oblique, sur plusieurs axes à la fois. Le QR code apparaît donc en perspective (cf. image). En outre, l'image est généralement déformée du fait même de la lentille optique du capteur. Cette dernière est conçue pour prendre des photos à une distance donnée. Les clichés pris de plus près ou plus

loin font donc apparaître des distorsions, soit vers l'extérieur (cf. image) soit vers l'intérieur.



Déformation oblique lors du scan



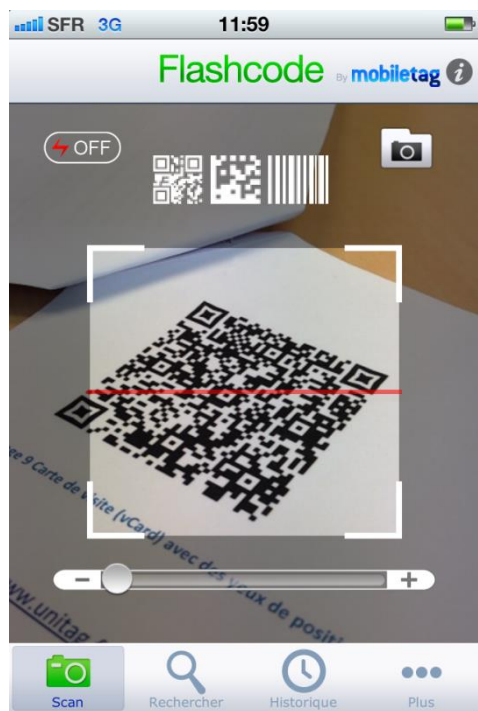
Alignement



Distorsion de l'objectif

Les yeux de positionnement et d'alignement servent à compenser ces déformations de l'image. Il est donc important qu'ils soient lisibles. Cela implique de ne pas en changer la fonction lors de la phase de personnalisation mais surtout que le support d'impression soit le plus plane possible. La pire situation est quand le code est imprimé sur un support courbe, il devient difficilement lisible. Ainsi on voit parfois des flyers collés sur les poteaux téléphoniques ou les abris de bus. Ceux-ci étant ronds, le QR code devient quasi illisible. Le logiciel

doit alors d'abord utiliser de puissants algorithmes pour redresser l'image.



Scan oblique en cours sur iPhone

En outre, les scanners imposent une distance minimale pour réaliser le focus (la mise au point de l'image). Les smartphones de dernières générations disposent tous d'autofocus assez performants mais il ne faut pas oublier les utilisateurs qui ne changent pas de matériel chaque année. Et c'est sans compter la qualité du capteur utilisé... Cela implique que le QR code soit imprimé dans une taille assez importante et qui dépend directement de la distance du sujet. Par exemple, un QR code imprimé sur une affiche sera scanné de plus loin que le même code dans un magazine. Il devra donc être imprimé plus gros.



QR code design d'Unitag

Unitag propose un guide des 10 commandements du QR code sous la forme d'un petit livret en téléchargement libre sur son site. On y découvre notamment l'impact du niveau de qualité sur l'image finale, sa lisibilité sur un support donné, les personnalisations les plus adaptées selon le contexte, etc.

Le livret présente également des exemples concrets de campagnes marketing réussies ou ratées. On comprend ainsi que l'ajout d'un QR code sur des

publicités affichées dans le métro et renvoyant vers le site Web du fabricant n'a que très peu d'intérêt dans un espace clos où le Wifi (et donc la connexion Internet) est quasi absent. A l'opposé, le livret insiste sur l'importance de proposer des contenus de qualité derrière le QR code, faute de quoi son intérêt est limité.

La plupart des smartphones disposent d'applications de lecture des QR codes, soit de façon native soit à partir de leurs « app stores » respectifs. En plus de bloquer régulièrement à propos des derniers QR codes qui valent le coup d'œil, le site QR Dress Code présente une sélection des meilleurs générateurs de QR code en ligne, classés par fonctionnalités, et des scanners les plus performants.

Un dernier point à garder en tête est le type de scanner utilisé. Ainsi, quand le QR code renvoie vers un site Web, et que le lecteur cible est un iPhone, il faut que le site soit consultable depuis le terminal. Cela implique de ne pas utiliser la techno Adobe Flash (illisible sur iPhone) et d'avoir une présentation adaptée au format (petit) de l'écran.



Thierry Leriche-Dessirier
Architecte JEE freelance
<http://www.icauda.com>

Carte de visite (vCard) avec
des yeux de position en croix

Unitag : <http://www.unitag.fr>

QR Dress Code : <http://www.qrdresscode.com/>

Cet article a initialement été écrit pour le magazine Programmez et a été publié dans le numéro 157 de novembre 2012.