

# Projet – Lecture de QR code

Version du 28 février 2022

Yann Strozecki – [Yann.Strozecki@uvsq.fr](mailto:Yann.Strozecki@uvsq.fr)

Le but de ce projet est de programmer un lecteur de QR code. Nous implémenterons une version simplifiée, mais très proche de la réalité. Pour vous renseigner sur les QR codes et leur fonctionnement, regarder la vidéo <https://www.youtube.com/watch?v=wKB-ZI2gXsk> et la page wikipedia [https://fr.wikipedia.org/wiki/Code\\_QR](https://fr.wikipedia.org/wiki/Code_QR).

Le projet est à faire en Python, par groupe de 2. Vous présenterez votre travail lors d'une soutenance la semaine du 16 mai, sur les horaires de TD habituels. Des exemples de QR codes sont donnés avec ce document, pour tester si votre code fonctionne.

## 1 Lecture d'un QR code

On considère uniquement des QR codes de taille  $25 \times 25$ . Le coin en haut à gauche est de coordonnées  $(0, 0)$  et celui en bas à droite est de coordonnées  $(24, 24)$  (lecture de gauche à droite et de haut vers le bas, comme c'est classique pour les images). Ils sont donnés comme des images en noir et blanc et on vous fournit le code qui permet de charger et sauvegarder une image en noir et blanc dans le fichier *projet.py*.

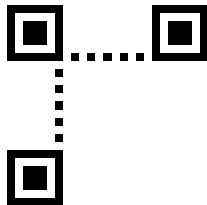


FIGURE 1 – Le squelette d'un QR code avec 3 symboles de coin et les lignes pointillées

**Question 1 :** Un symbole de carré noir de 3 pixels sur 3, entouré d'une bande blanche, entouré d'un bande noire. Enfin, une bande blanche est positionnée sur les deux côtés orientés vers l'intérieur de l'image. Voir ce symbole dans la figure 1. On retrouve cette figure dans tous les coins sauf dans le coin inférieur droit d'un QR code. Donner une fonction qui génère une matrice représentant ce symbole. Programmer une fonction qui trouve dans quel coin ce symbole n'apparaît pas et applique une rotation à l'image pour la positionner correctement.

**Question 2 :** Des lignes qui alternent pixel noir et pixel blanc rejoignent les symboles des coins de l'image, voir la figure 1. Écrire une fonction qui vérifie que ces lignes apparaissent correctement.

**Question 3 :** Les informations sont codées avec un code de correction d'erreur de Hamming. Programmer une fonction qui lit 7 bits (donnés comme une liste de 0 et de 1) et qui renvoie les 4 bits d'information obtenus après correction d'erreurs. Les codes de Hamming sont stockés avec d'abord les 4 bits de message, puis les 3 bits de parité, comme expliqué dans le cours (le codage de Hamming standard utilise un entrelacement des bits plus compliqué). Dans les vrais QR codes, le code de correction d'erreur utilisé est le code de

Reed-Solomon, plus sophistiqué et les bits de corrections sont stockés séparément.

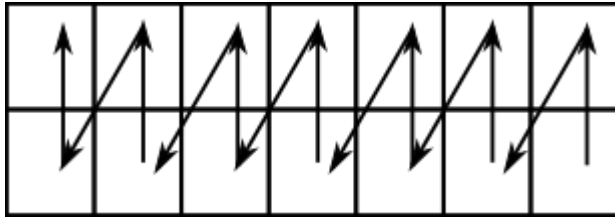


FIGURE 2 – Le parcours d'un bloc de taille 14 de la droite vers la gauche

**Question 4 :** Les informations se lisent par bloc de 14 bits qui contiennent deux fois 7 bits codés avec le code de Hamming(7,4).

Le sens de lecture à l'intérieur d'un bloc est donné dans la figure 2. La lecture des blocs commence par le pixel en bas à droite du QR code. On lit deux blocs de la droite vers la gauche, puis en montant de deux pixels dans la figure, deux blocs de la gauche vers la droite et ainsi de suite en lisant au plus 8 séries de 2 blocs. Dans les vrais QR codes, il peut y en avoir plus et leur placement est légèrement différent.

Programmer une fonction qui parcourt l'image d'un QR code pour renvoyer l'information lue sous la forme d'une liste de listes de 14 bits.

**Question 5 :** Un bloc contient 8 bits d'information. Pour interpréter les données d'un bloc, il faut savoir ce qu'elles représentent. Le pixel à la position (24, 8) code le type de données. S'il est à 0 il s'agit de données numériques : 8 bits codent deux symboles hexadécimaux. S'il est à 1, il s'agit de données brutes, 8 bits seront interprétés comme un code ASCII. Les codes ASCII et les symboles hexadécimaux sont stockés en big endian, c'est à dire avec le bit de poids faible au début et le bit de poids fort à la fin (comme on a vu dans le premier TD et comme on écrit les nombres à la main). Pour simplifier, nous supposons que les symboles hexadécimaux d'un message sont toujours en nombre pair, ainsi les blocs seront toujours entièrement remplis.

Afficher le contenu d'un QR code, en prenant en compte le type de données.

**Question 6 :** Pour rendre l'image plus lisible, un filtre est appliqué à l'image. Ce type est stocké sur deux bits de contrôle dans les pixels (22, 8) et (23, 8). Le filtre est appliqué en effectuant un XOR entre les pixels du filtre et ceux du QR code. Les images de filtre sont :

- 00 entièrement noire (pas de changement) ;
- 01 un damier dont la case en haut à gauche est noire ;
- 10 des lignes horizontales alternées noires et blanches, la plus haute étant noire ;
- 11 des lignes verticales alternées noires et blanches, la plus à gauche étant noire.

Écrire une fonction qui applique le bon filtre selon les bits de contrôle.

**Question 7 :** Les informations à stocker dans le QR code peuvent nécessiter moins que les 16 blocs disponibles. Nous allons stocker le nombre de blocs à décoder en base 2 sur 5 bits dans les pixels (13, 0), (14, 0), (15, 0), (16, 0) et (17, 0). Quand vous décoderez un QR code, vous calculerez d'abord le nombre de blocs utilisés et vous l'afficherez puis vous décoderez uniquement ces blocs.

Décoder tous les QR codes donnés en exemple, pour vérifier si votre code fonctionne. Le QR code contient une suite de valeurs hexadécimales qui commence par 14BAD et contient 10 valeurs.

## 2 Écriture de QR code

Cette section est complètement facultative, et destinée à ceux qui veulent aller plus loin. Il s'agit de réaliser le travail inverse, à partir d'un message donné, créer l'image du QR code qui contient ce message.

### 3 Modalités pratiques

Merci de respecter les consignes suivantes :

- le projet est à faire en binôme ;
- le projet est à rendre dans MOODLE, au plus tard le **lundi 16 mai à 9h00** ;
- une soutenance aura lieu la semaine du 16 mai pendant votre séance de TD.
- vous devez déposer un fichier `XY_NOM1_Prenom1-NOM2_Prenom2.zip` qui est le zip du dossier `XY_NOM1_Prenom1-NOM2_Prenom2` contenant votre projet.  
**XY** sont les initiales de votre chargé de TD (LD, CG, MT).
- Un outil de détection de plagiat sera utilisé sur vos codes.

Le retard de la remise du projet entraîne 1 point de moins par heure de retard.