SmartCode – Projet QR Code

Documentation

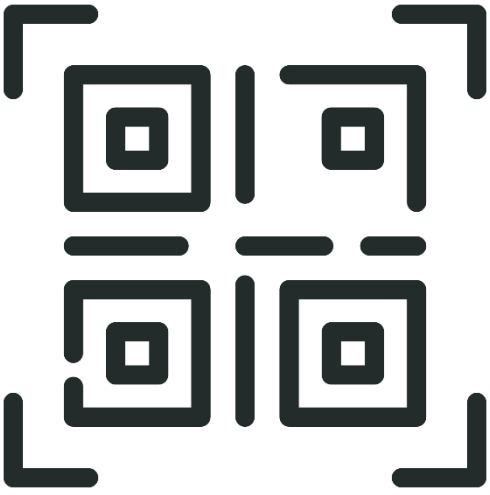


Table des matières

Introduction....3

Présentation....3

Vues....4

Spécifications....5

Installation....6

Introduction

Projet effectué dans le cadre du cours « Reconnaissance d'Images avec Application sur Mobile » du semestre 2 au sein du Master 1 Informatique et Mobilité à l'Université de Haute-Alsace.

Le groupe de projet est composé de :

- SEYLLER Vincent,
- DALLEST Bryan.

Le but était de développer une application mobile permettant de générer son propre QR Code à partir d'un champ de texte, et de pouvoir le décoder. Ce QR Code devait être spécifique à l'application, utilisant des techniques de codage / décodage propre à elle-même.

Projet GIT de l'application : https://github.com/DLTBryan/SmartCode.

Présentation

L'application a été développée sous Cordova[1], permettant de faire du JS sur mobile. La seule librairie utilisée a été OpenCV[2] pour la récupération du QR Code depuis la caméra. L'encodage et le décodage ont été fait en JS natif avec l'utilisation de Canvas.

[1] https://cordova.apache.org/
[2] https://docs.opencv.org/3.4/d5/d10/tutorial_js_root.html

Vues

Voici les différentes vues de l'application :

Page d'accueil

Page d'encodage

Page de résultat d'encodage

Page de partage

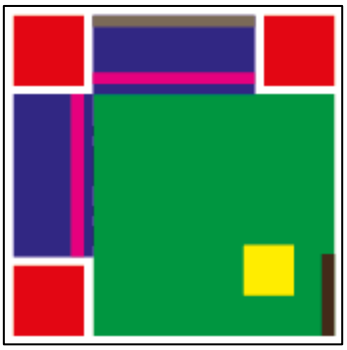
Page de décodage

Page de résultat

Spécifications

Voici un exemple de QR Code généré par notre application :

Voici les différentes zones du QR Code :



- Rouge : Marqueur de positions
- Bleu : Partie générée aléatoirement
- Gris : Type du QR Code
- Rose : Timing pattern
- Jaune : Marqueur d'alignement
- Vert : Partie permettant de stocker la donnée
- Brun : Partie contenant le caractère de début

La donnée stockée peut aller jusqu'à 67 caractères ASCII. Le début de la donnée commence forcément par le caractère de début codé en ASCII (défini à * dans notre code afin d'éviter la distorsion de l'image lors de la détection). Si jamais la donnée est plus petite, le caractère de fin de mot (EXT en ASCII) est ajouté à la fin du mot et le reste de la zone verte sera généré aléatoirement. Pour le stockage de la donnée, le QR Code est lu de bas en haut dans la zone verte jusqu'à arriver au caractère de fin de mot ou à la fin de la zone verte.

L'ensemble du QR Code (sauf les marqueurs de position, les timing pattern et le marqueur d'alignement) est chiffré via un masque, et la détection fonctionne de la manière suivante :

- L'utilisateur détecte un QR Code avec la caméra,
- Le QR Code est déchiffré (le masque est appliqué),
- On vérifie l'intégrité du QR Code : l'alignement des marqueurs de position, du marqueur d'alignement et le caractère de début de mot,
- Si le QR Code est intègre et que le type correspond au type du QR Code de l'application (ici « SD »), on déchiffre l'ensemble du QR Code et on affiche la donnée décodée,
- Sinon on ne fait rien.

Cela permet de décoder uniquement les QR Codes correspondants à l'application.

Chaque utilisateur peut générer lui-même son QR Code et le partager via la fenêtre « Share » native de son téléphone Android (Cf. Vues).

Installation

Pour l'installation, il suffit de télécharger l'APK disponible via les différentes releases disponibles sur le dépôt GIT du projet : https://github.com/DLTBryan/SmartCode/releases.

Il ne faut pas oublier d'accorder l'accès à la caméra lors du premier lancement de l'application.

La détection ainsi que le décodage du QR Code peut être assez longue, étant donné que l'application prend par défaut la caméra 1 du téléphone, qui peut être un grand angle et donc générer une grosse distorsion de l'image, ce qui complique le déchiffrage. Sur ordinateur, la détection est presque instantanée.