

Réalité augmentée sur téléphones, tablettes et lunettes

UE : Projet de fin d'études
Master 2 Informatique pour
l'image et le son
Client : Pascal DESBARATS

Domaine / Sujet



Source : phoneandroid.com



Source : EPSON

Application de réalité augmentée :

- sur smartphone / tablette Android [1]
- sur lunettes Epson Moverio BT-300 [5]

But : afficher des modèles (2D ou 3D) sur des marqueurs et des modèles 2D sur des feuilles blanches.

Domaine / Sujet



Source : fr.wikipedia.org



Source : argentwebmarketing.com

Deux applications, deux buts :

- Détecter des QR codes pour afficher des modèles 3D dessus
- Détecter une feuille blanche pour afficher une image par dessus.

Au final, trois applications :

- QR code sur téléphone / tablette.
- Feuille blanche sur téléphone / tablette.
- Feuille blanche sur lunettes.

Application : Feuille blanche



- Utilisation d'Android Studio [7].
- Utilisation de la bibliothèque OpenCV [2] pour détecter la feuille blanche.
- Utilisation du *Software Development Kit* de EPSON pour les Moverio BT-300 [8].



Source : supinfo.com



Source : opencv.org



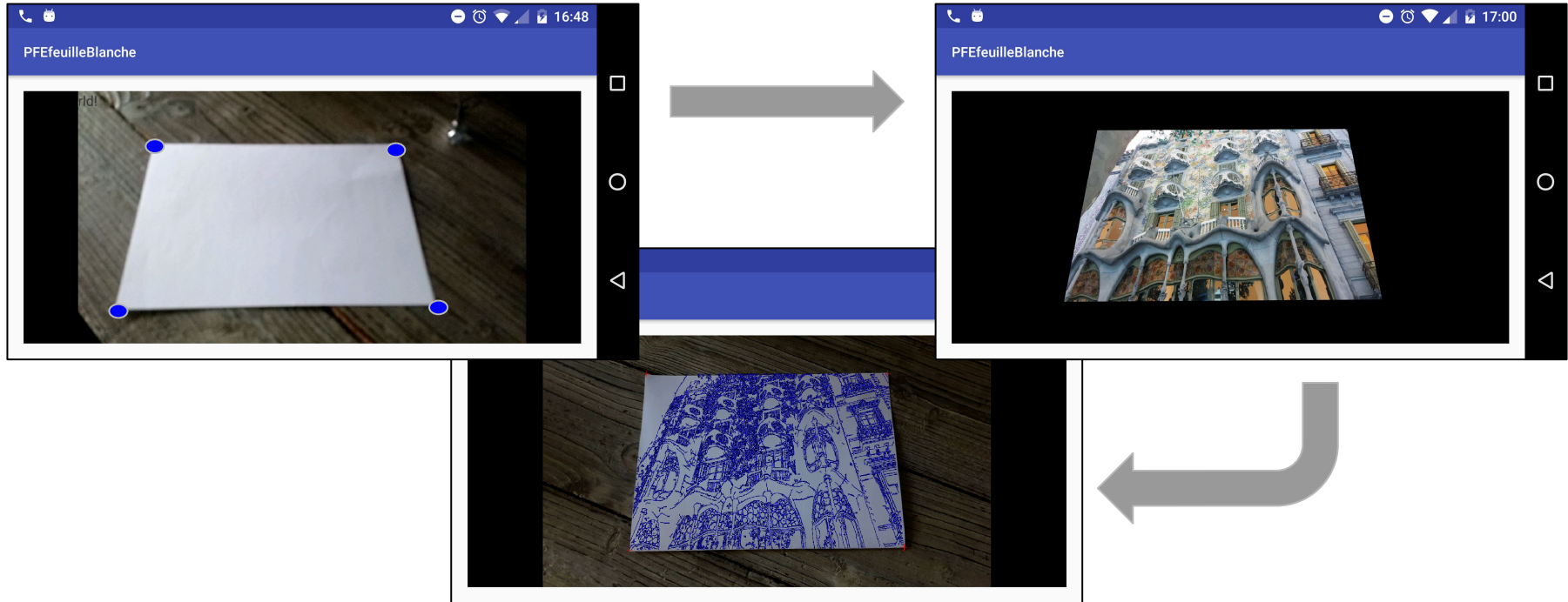
Source : EPSON



Source : EPSON

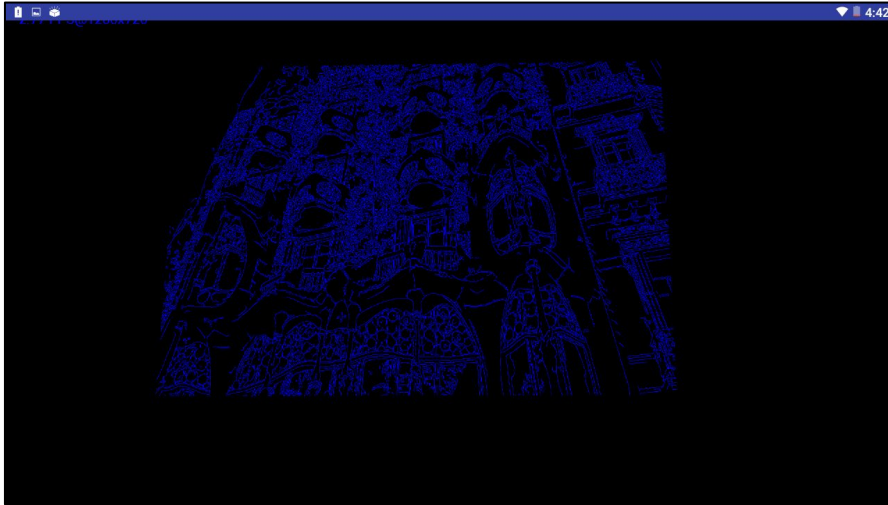
Application : Feuille blanche

Téléphone / tablette



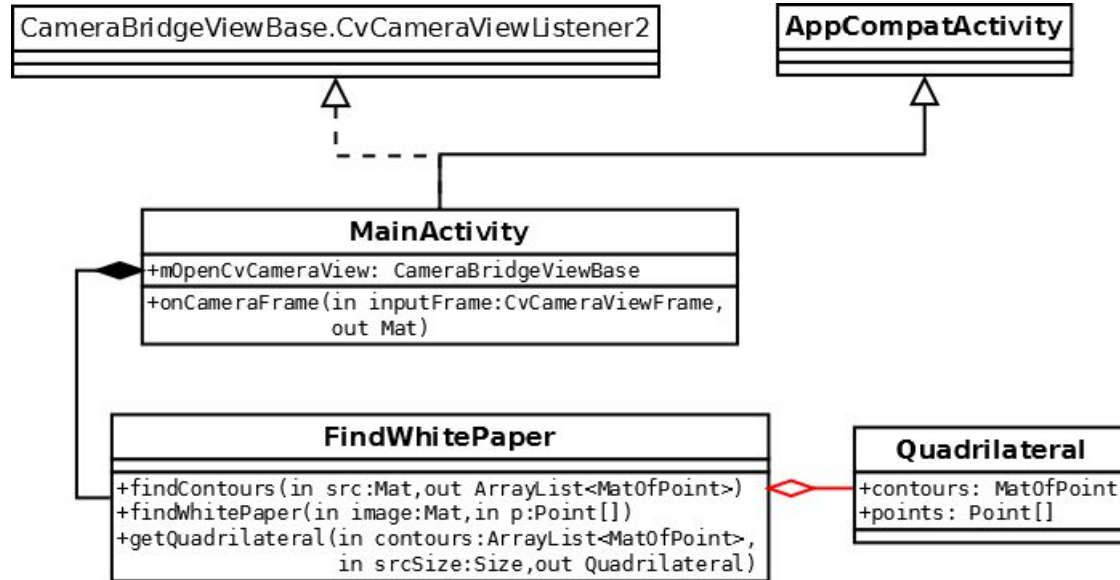
Application : Feuille blanche





Lunettes



- Interface minimaliste
 - Pas de barre de navigation
 - Pas de barre d'outils
 - Barre de notification

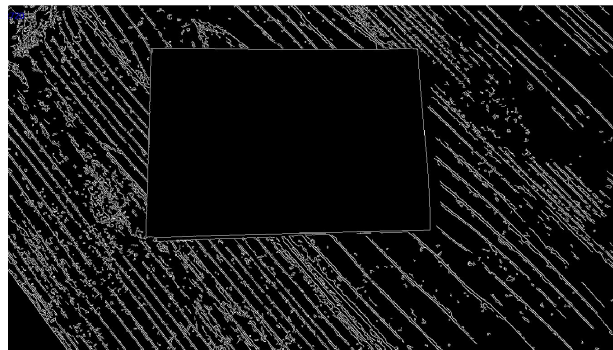
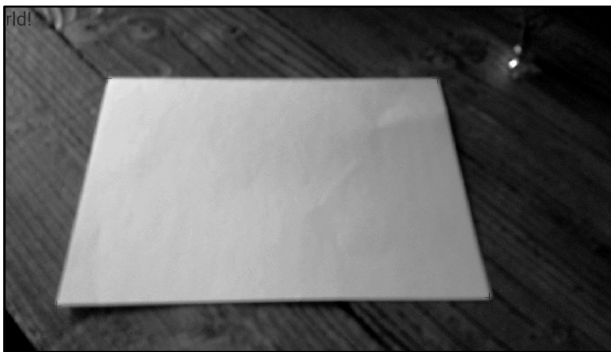
Architecture : Feuille blanche



- A  B : B implémente A
- A  B : B hérite de A
- A  B : B compose A
- A  B : B est une classe interne à A

Points techniques : Feuille blanche

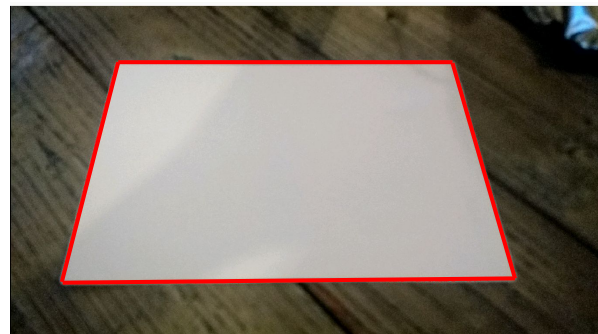
Détection d'une feuille blanche



- Passage en niveaux de gris
- Filtre de Canny

Points techniques : Feuille blanche

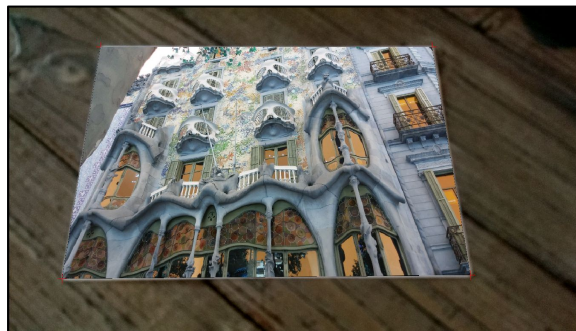
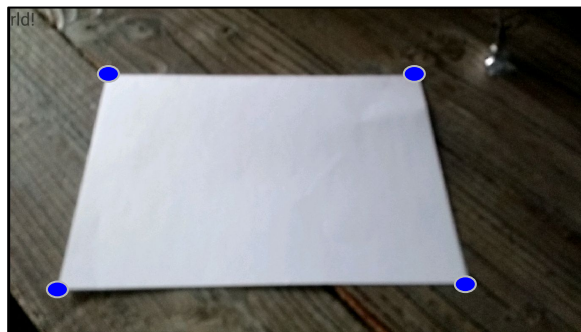
Détection d'une feuille blanche



- Détection des contours
- Approche du plus grand contour par un polygone
 - code d'une application de Scanner

Points techniques : Feuille blanche

Détection d'une feuille blanche



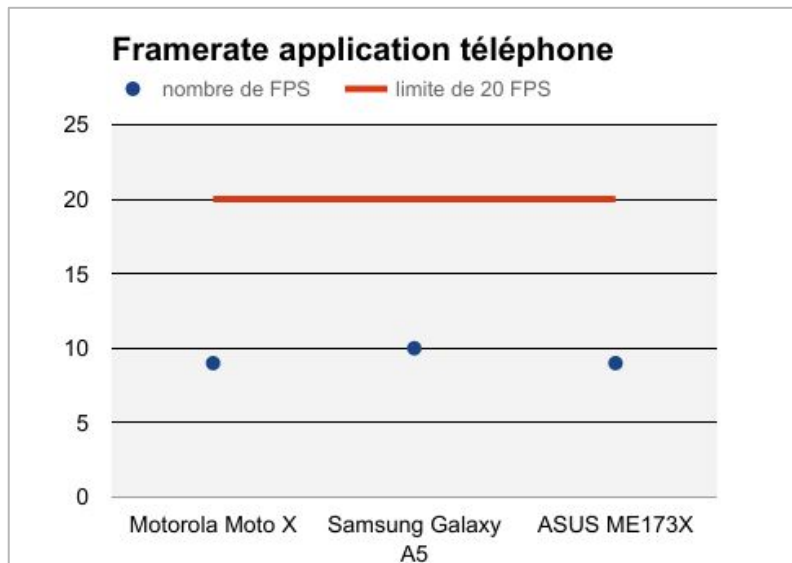
- Détection des coins du polygone
- Calcul de la transformation entre l'image à afficher et la projection de la feuille
- Appliquer la déformation à la feuille

Bilan : Tests

Feuille Blanche



- Téléphone / tablette :
 - FPS



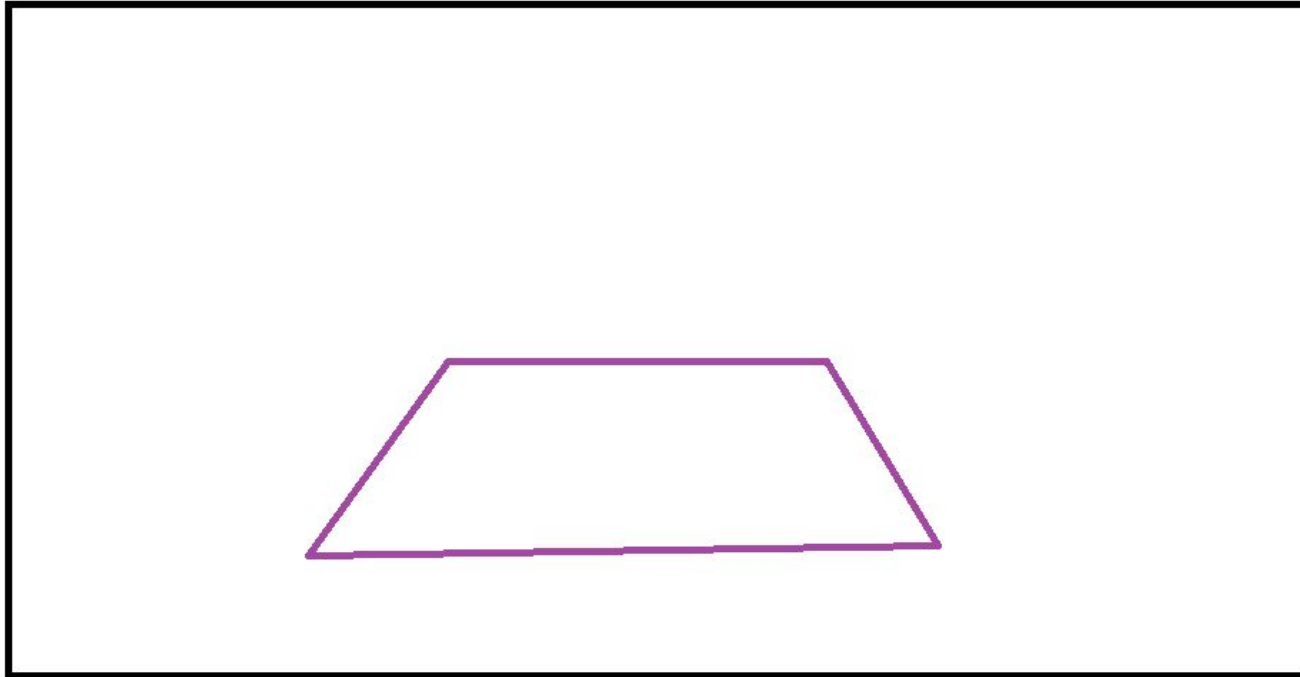
- différents formats de papier : A4 à A7.
- luminosité : sensible aux changements de lumière
- visuellement acceptable sur téléphone

- Lunettes

- FPS : 7
- luminosité : sensible
- visuellement acceptable : pas pour le moment

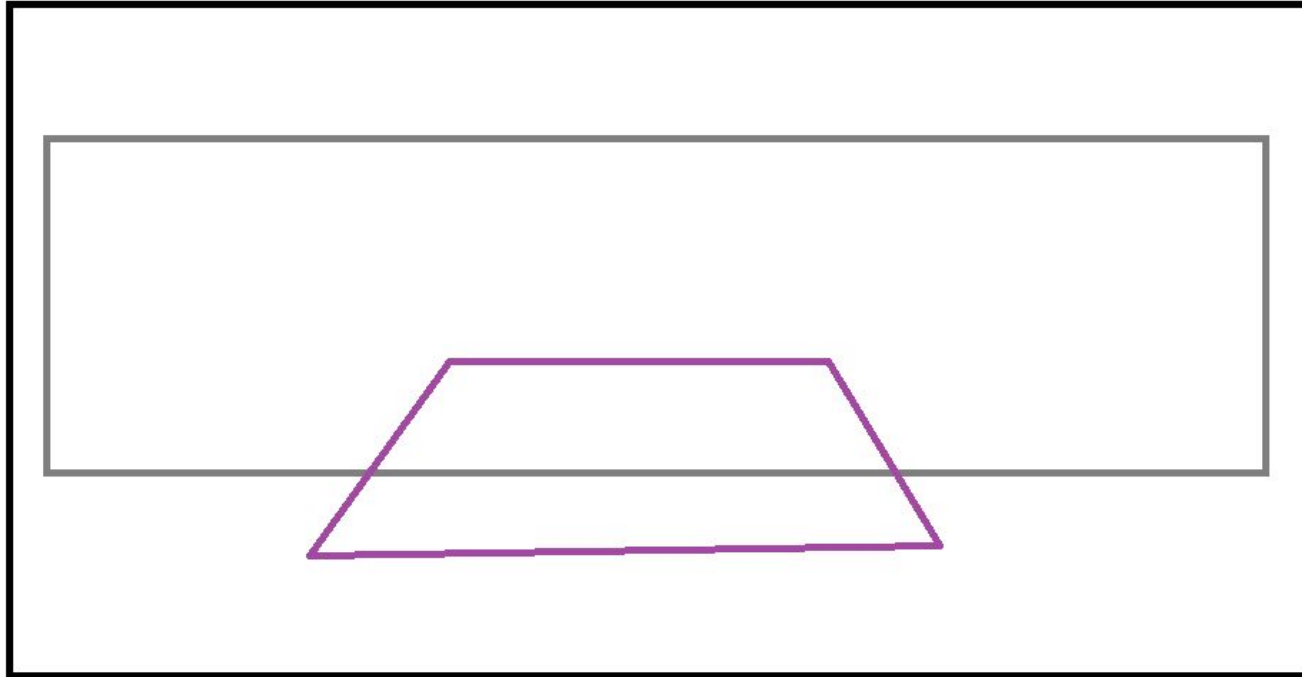
Bilan : Tests

Feuille Blanche



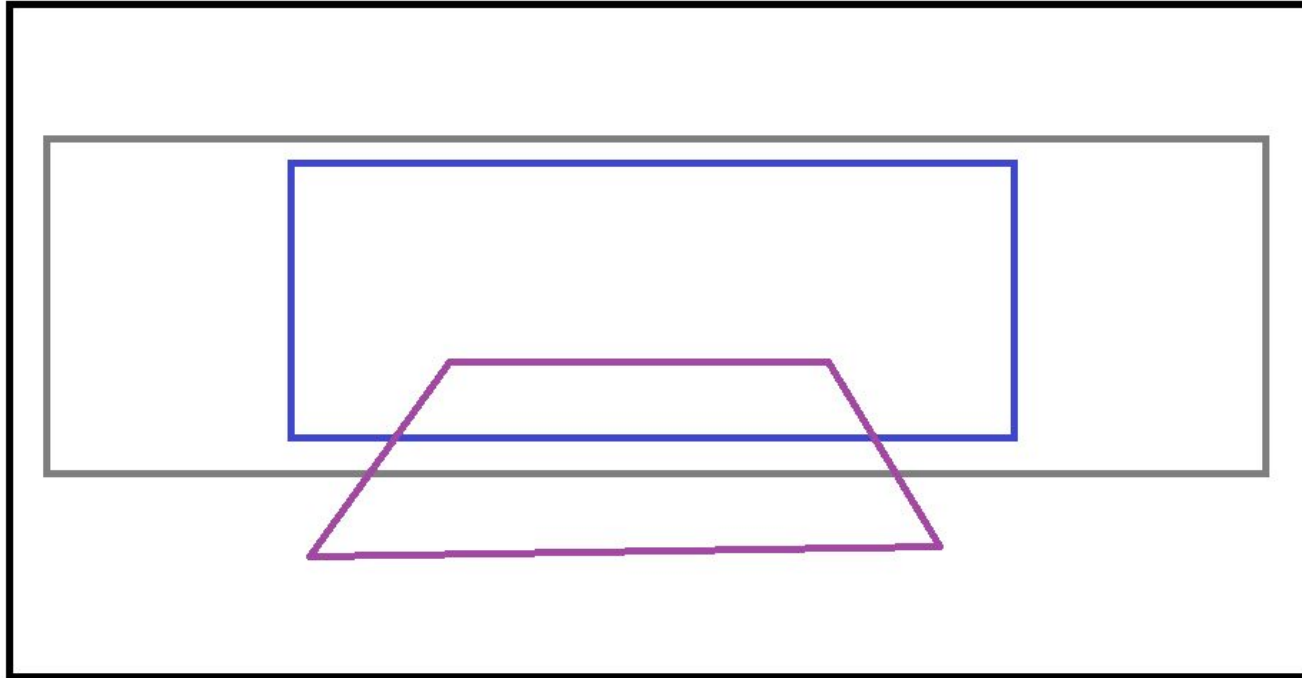
Bilan : Tests

Feuille Blanche



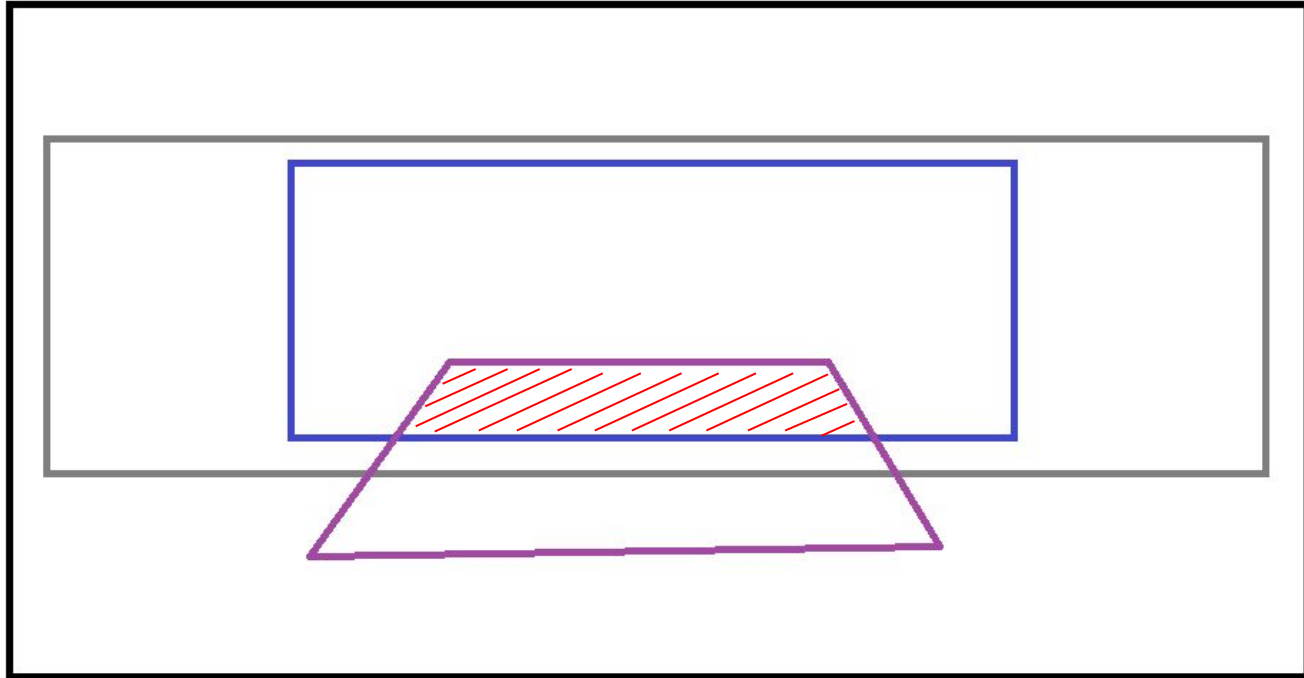
Bilan : Tests

Feuille Blanche



Bilan : Tests

Feuille Blanche



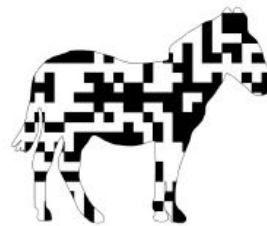
Application : QR Codes sur téléphone / tablette



- OpenCV [2] → détection et recalage de QR Codes.
- Lecture du contenu avec Zxing [3].
- Affichage d'un modèle 3D par-dessus le QR Code.
- Application hybride en C++ et Java grâce à la JNI (*Java Native Interface*).



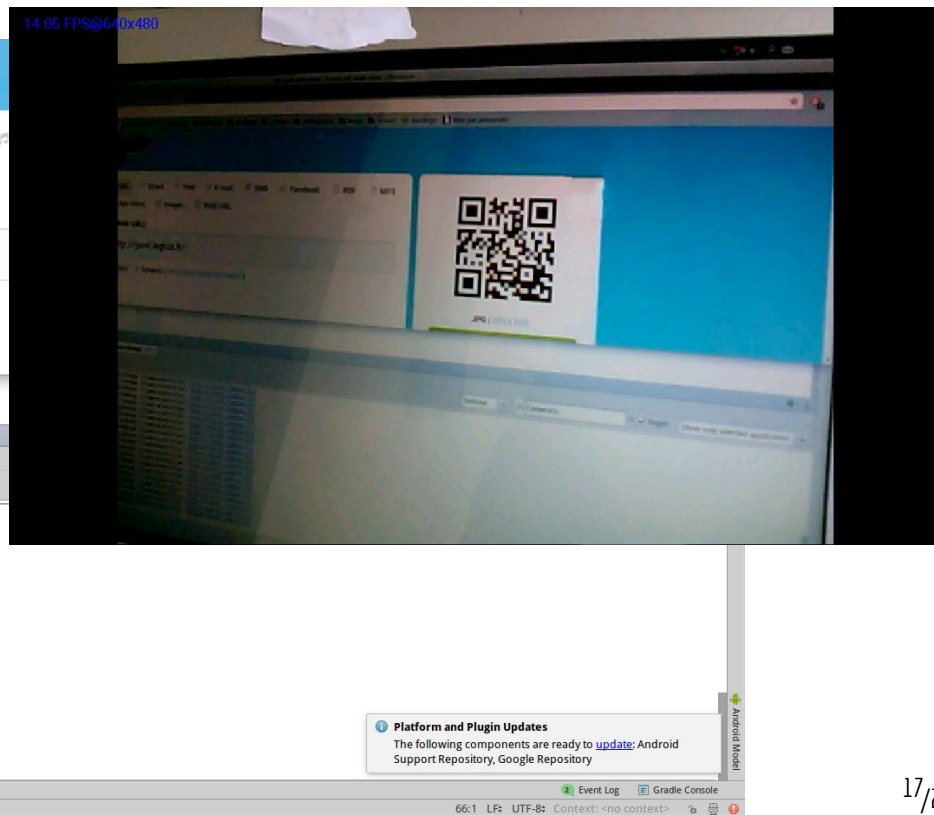
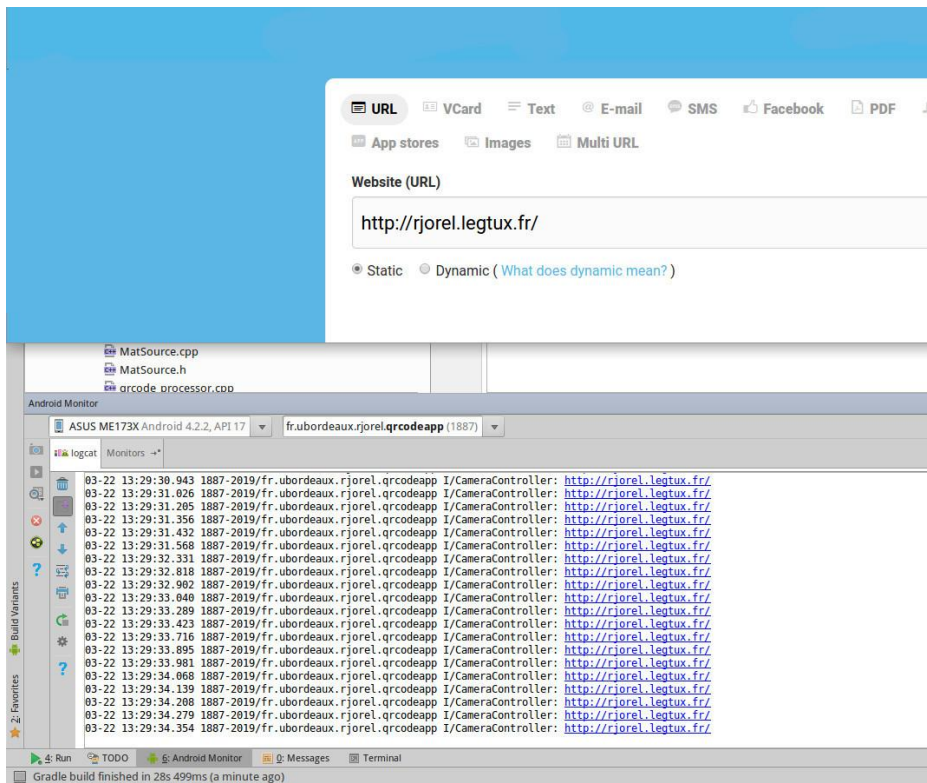
Source : <http://opencv.org/>



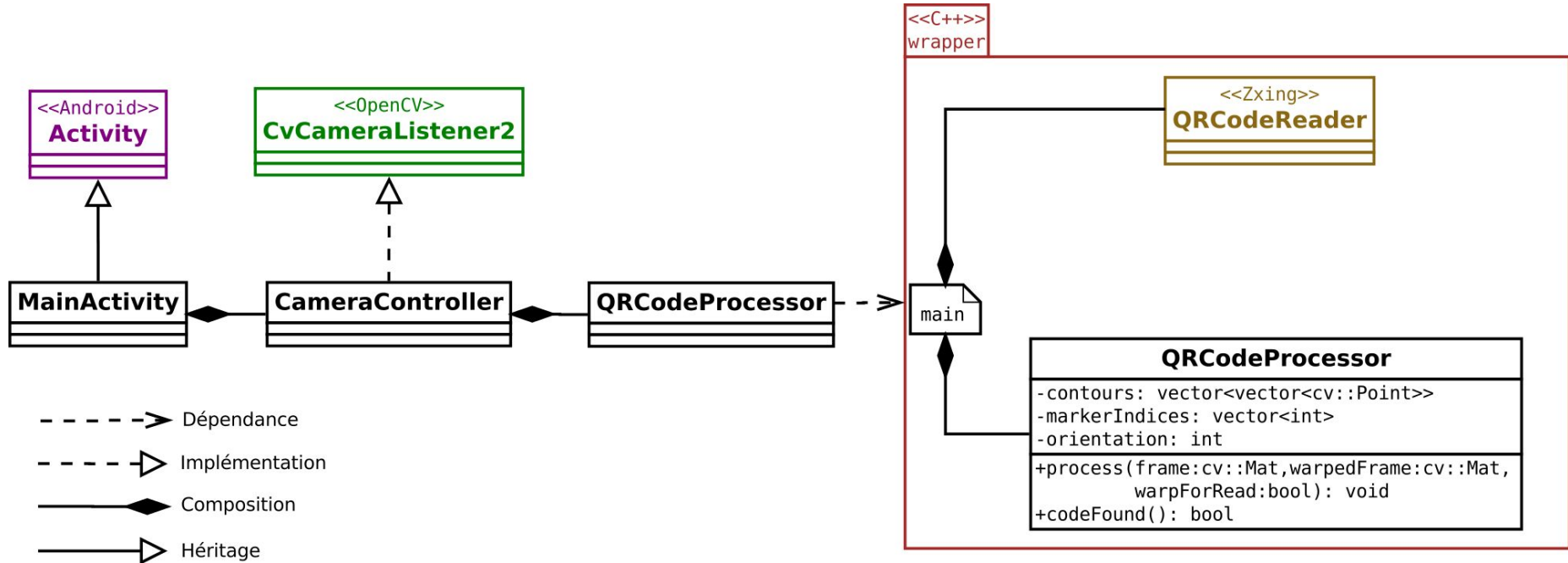
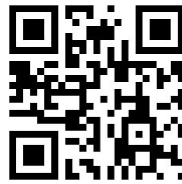
ZXING

Source : <https://zxing.org/w/decode.aspx>

Application : QR Codes sur téléphone / tablette

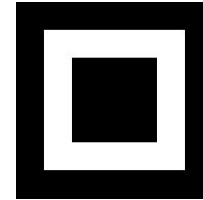


Architecture : QR Codes



Points techniques : QR Codes

Principe



Finder pattern

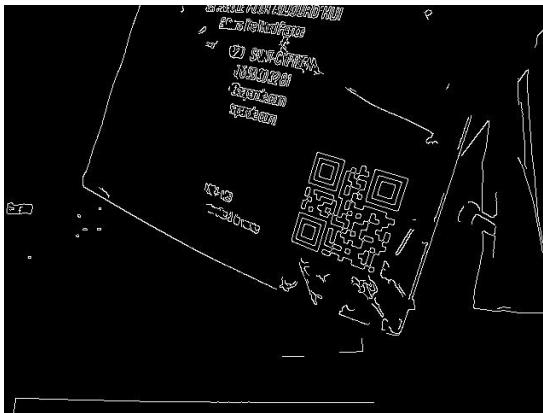


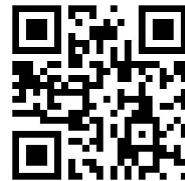


Points techniques : QR Codes

Algorithme de détection [4]

- Détection des *finder patterns*
 - ◆ Canny + calcul des contours,
 - ◆ Parcours des contours pour trouver les *finder patterns*
- Réorganisation des patterns + orientation





Points techniques : QR Codes

Algorithme de détection [4]

- Recalage du QR Code
 - ◆ Calcul des bords du QR Code
 - ◆ *Warping*
- Lecture du QR Code avec Zxing





Bilan : Tests

QR codes

- *Framerate.*
 - 10 - 15 FPS en moyenne sur Asus ME173 X (Processeur : MTK MT8125 / 8389 1.5 GHz, RAM : 1024 Mo).
- Taux de reconnaissance + lecture : ~10 %.
- Taille minimale des QR Codes : 1.5mm / pixel, à 30cm du QR Code.
- Luminosité : importante sensibilité à la lumière.



Bilan : Besoins

QR codes

- ~ Faire une application android pour smartphone et tablette :
 - ✓ Récupérer le flux vidéo d'un terminal Android,
 - ✓ Détecter un QR Code,
 - ✓ Lire son contenu,
 - ✗ Afficher modèle 3D par-dessus.
- ✗ Adapter cette première application aux lunettes Epson.



Bilan : Besoins

Feuille blanche

- Faire une application android pour smartphone et tablette :
 - ✓ Afficher une image 2D sur une feuille blanche.
- Adapter cette application aux lunettes Epson :
 - ✓ Ne pas afficher le flux vidéo.
 - ~ Prendre en compte le décalage.
 - ✗ Faire de la stéréoscopie.
 - ✗ Faire une interface graphique.

Bilan : Besoins

Non fonctionnels

- mouvement en temps réel
 - ✗ 20 FPS → natif ?
- réutilisabilité / généricité
 - ✓ briques de base portables
 - ~ une documentation
 - ✓ modules sans effets de bords
- fiabilité
 - ~ stabilité
- précision
 - ~ placement visuellement acceptable des modèles 2D

Bilan : Perspectives

QR code :

- Afficher un modèle 3D sur un QR Code, en fonction du contenu.
- Améliorer la détection et le recalage pour la lecture.

Feuille Blanche :

- Charger une image du téléphone ou la prendre en instantané.
- Passer en natif pour augmenter le nombre de FPS.
- Faire de la stéréoscopie.
- Améliorer la robustesse de la détection de la feuille blanche.

Références

- [1] - <https://developer.android.com/index.html>
- [2] - <http://opencv.org/>
- [3] - <https://github.com/zxing/zxing>
- [4] - <http://dsynflo.blogspot.fr/2014/10/opencv-qr-code-detection-and-extraction.html>
- [5] - <https://www.epson.fr/products/see-through-mobile-viewer/moverio-bt-300>
- [6] - <https://github.com/ctodobom/OpenNoteScanner/>
- [7] - <https://developer.android.com/studio/index.html>
- [8] - https://tech.moverio.epson.com/en/bt-300/sdk_download.html

Réalité augmentée sur téléphones, tablettes et lunettes

UE : Projet de fin d'études
Master 2 Informatique pour
l'image et le son
Client : Pascal DESBARATS